Universidad Carlos III de Madrid

Sistema de voto electrónico mediante autenticación con DNIe

TRABAJO DE FIN DE GRADO EN INGENIERÍA INFORMÁTICA

JUNIO DE 2017

Tutor: José María Álvarez Rodríguez **Email:** joalvare@inf.uc3m.es

Alumno: Raúl Bejarano Parrilla **Email:** 100303632@alumnos.uc3m.es

Agradecimientos

A lo largo de la carrera he conocido a mucha gente. Primero erais solo compañeros, pero después os convertisteis en amigos. Juntos hemos pasado buenos y malos momentos, momentos de nervios, de frustración, de desesperación, pero sobretodo de risas, charlas entretenidas, bromas y alegrías. Es a vosotros a quien doy gracias por estos momentos, que espero podamos repetir muchas veces en el futuro.

Gracias a mis profesores de la universidad y el instituto. En especial a Carlos Toledo y Elena Cedenilla, que hicieron que me picara la curiosidad por la ciencia y la tecnología. Gracias a todos aquellos profesores que viven con pasión su oficio e inculcan esa pasión en sus alumnos.

Gracias a ti, Cristina, por aguantar mis rarezas, mis agobios, mi parsimonia y mi efusividad. Gracias por hacerme mejor persona y querer ser mi compañera de viaje en este viaje que es la vida.

Para el final he dejado a los más importantes, mi familia. A vosotros os lo debo todo. Gracias por apoyarme, por ayudarme a levantarme cuando me caigo, por darme oportunidades que quizás no mereciera y de las que, a veces, no soy consciente de vuestro sacrificio. Gracias de corazón.

"Siendo ciudadano de un Estado libre, por poca influencia que pueda tener en los asuntos públicos, el derecho a votar basta para que tenga el deber de instruirme"

Jean Jeaques Rousseau

SUMMARY

Introduction

Election by vote is on the basis of a democratic system and, therefore, key in our society. The technology, which has provided great progresses in other subjects, however, has not yet managed to offer a viable alternative to this type of traditional systems.

Historically, Information Technologies have proposed solutions to implement voting systems. One of the problems encountered is the variety of voting models that make little synergy between them (and therefore costlier). Even so, there are certain aspects of all these systems that present common points, such as voter authentication systems and subsequent validation of votes. These are aspects that will focus this End of Grade Work (FDP).

The main requirements of these systems are that they must be very safe, accessible for all, auditable and closely related to the latter, open source. Safety is a must due to the fact that the handled data are personal and protected by the Organic Law of Data Protection [1] as very sensitive data to contain information on political affiliations. They must be accessible so that any citizen can use them regardless of their physical condition and age. Auditable by external and independent entities that verify the result of an election in a clear way and that does not give rise to doubts of the veracity of the vote. If they are open source, it also allows auditors to know what the system does and provide a fundamental transparency.

There is a case of success of electronic voting which is considered a world reference, this is the case of Estonia where citizens can vote online using a Spanish like identity card.

Digital identity is the trail left by a user through interaction with digital services. As Internet users, our digital trail is quite evident every time we register on websites, upload personal content, express our opinion, etc. This is easily verifiable with a simple search of our name in any search engine [2], [3].

The problem comes when checking if information is valid and provided by the user who uses these digital services. The solution that most companies have taken is based on the information that other providers offer, see Facebook or Google that allow the user authentication through their services, preventing users from having to create new identities in each service they want to use.

These systems have two major problems: the first is that they do not provide a mechanism to verify that a personal identity is unique, that is, a person can have multiple accounts in one of these services; The second is that these providers do not offer a unified service and therefore end up creating multiple identities depending on the provider that the user chooses when authenticating on another web.

There are, of course, other viable alternatives, but are not spread such as. "OpenID is a digital identification standard in which a user can be identified in a web page through a URL and can be verified by any server

that supports the protocol" [4]. This solves the problem of unified services, but not that of the unique identity. In addition, it has received a lot of criticism regarding the security and privacy related to the confidentiality of the services consumed and that the commitment of the security of the user's device all its information could be unprotected.

Motivation

There is much controversy as to whether it is possible to carry out a voting system in Spain with the means at our disposal [5]. The motivation behind this Final Degree Project is to have a first contact with the technologies that would make possible the creation of these systems in accordance with the regulations in force in Spain and determine their feasibility when implementing them.

As a personal motivation, I must add that it is a project that I had in mind for a long time and this End of Grade Work offers me the possibility of dedicating a time and effort that in another situation could not dedicate.

Objectives

The main objective of this Final Degree Project (FDP) is to analyze, design and implement a voting system whose voter authentication and validation of votes is carried out through the use of the certificates included in the National Identity Document.

As an additional objective, I will try to study different electronic voting mechanisms and try different commercial solutions that offer this service. I will focus on how they solve the problem of digital identity and the verifiability of voting that they offer.

Voting over the Internet

Electronic voting can be defined as the process that allows an individual to exercise his vote in an electronic medium. It emerges as a tool to improve and facilitate voting access through the use of information technologies. Currently, there are multiple methods of voting, whether punch cards, voting with optical scanner, voting with certificates, etc. [6]. This Final Degree Project will focus exclusively on online voting systems.

Online voting systems provide many advantages to both organizing institutions and voters. For institutions, it mainly entails significant savings in spending, which involves mobilizing both material and economic resources when making any kind of voting. For voters, this is an improvement since they do not have to

travel until a traditional ballot box satisfies their voting rights [7]. We will discuss the advantages and disadvantages of these voting systems in more detail later.

Many of these voting systems use Smart-cards to perform voter identity verification. In Spain, we have the National Electronic Identity Document (DNIe) designed to adapt the old DNI to the information society and the consumption of electronic services. It began shipping in 2006 and has a cryptographic chip that provides authentication and signature certificates. This chip has digital security measures such as data encryption of the chip and access to internal functionality using a password known only to the citizen. Also, it should be noted that the keys never leave the chip. The technical aspects of this document will be detailed later.

Types of electronic voting

Before starting to review existing Internet voting systems, which is the subject of this FDP, I will begin by briefly explaining what types of electronic voting exist and what they consist of.

There are two large groups of electronic voting, physical voting and remote voting.

The physical vote is carried out in person in a voting center, in Spain, specifically in the polling stations. This type of voting entails supervision by electoral authorities that prevent coercion, fraud and other incidents from occurring. This type of electronic voting is carried out mainly with special voting machines where the voter chooses on a screen the option of his choice.

Remote voting means that the voter can vote remotely and without supervision. In addition, it has to adapt to the devices and mechanisms that the voters have at their disposal. In any case, this type of voting should always be accompanied by the physical voting model.

Advantages and disadvantages of Internet voting

In this section we will try to summarize and analyze the different advantages and disadvantages of these systems [10], [11].

Advantages:

- Cost. The cost of an online election would be greatly reduced.
- Allows you to vote from anywhere you have access to the Internet.
- It allows the voting to people with disability or difficulty of mobility to the electoral college.
- Improved mail voting that requires many resources and infrastructure.

- Environmental impact. It would avoid the use of unnecessary material resources, such as the role of ballots and lists.
- Statistics in real time. Participation and other non-counting data would be accessible from the outset.
- Speed of counting. It would be possible to know the result of the vote just after the time available.
- Encourages citizen participation.

Disadvantages:

- Security in these systems is the main disadvantage because relying on computer systems exposed to the Internet can be targeted attacks.
- Security in voting audit.
- Possible fraud by intruders.
- Access to vote by users not accustomed to the use of information technologies or unable to do so.
- Voting coercion.

As we can see, the advantages offered by Internet voting are more than the disadvantages. Even so, we must pay attention to these possible disadvantages as they directly affect the security, privacy and usability of these technologies.

Safety must be paramount in this type of system, as it affects data on political affiliation that are highly sensitive. That is why it is the biggest criticism that is made to Internet voting. Even so, there are already mechanisms and technologies that provide this security.

Security during voting auditing must be robust, i.e. the voting authority should not have access to the identity of the voter at the time the voter is able to see his or her vote. There are a multitude of solutions that respond to this problem, from centralized systems with very closed policies of access to information to decentralized systems and free software like smart-contracts in Blockchain networks.

In the event of possible system intrusion fraud, i.e. attacks by cybercriminals with the aim of modifying voting results, there are also several solutions. These range from the classic control of networks, system permissions to the previously mentioned Blockchain networks that do not need trust between their nodes. All this, together with a good planning and engineering of the security from the beginning of the developments, makes these attacks are unlikely and in case of being given that the damages caused are minimum and rectifiable.

The arguments about the insecurity of these systems are well founded, but through a correct design, engineering, development and security protocols can be alleviated, and even eradicate these vectors of attack.

As for accessibility to online voting mechanisms, there are mechanisms such as the use of ATMs or volunteers that either by phone call or visit in person allow to vote for the elderly, disabled and anyone who cannot or knows Using new technologies.

The comparison between advantages and disadvantages makes it clear that the advantages offered by these systems clearly benefit our democratic societies and that the disadvantages they entail are above all of a technical nature and have a solution.

Scope of the system

We can define the Assembly as an electronic voting system. Its objective will be to facilitate the collection, counting and viewing of votes in a typology of direct democracy oriented to all types of social groups. Specifically, users can create an account in the application with their National Identity Document (DNI), create assemblies with which to organize with other users and vote proposals using the electronic signature with the DNIe. The complete functionality will be defined throughout this section.

Assemblies

The system organizes citizens in virtual assemblies within their area of action. The area of action of a citizen corresponds to the information of the census included in his DNI, i.e. if a citizen is from Toledo its areas of action will be Toledo, the province of Toledo, the Autonomous Community of Castilla La Mancha and Spain.

We can define virtual assembly as the non-physical place where people gather to give their opinion on common issues. An assembly can be constituted in any organization, company, community of neighbors, equipment of soccer, etc.

In addition, the assemblies may be public or private. The first will be accessible to everyone, anyone who wants to be a member can be. The latter, however, are not accessible to everyone, and will depend on the administrator (who is subject to the same selection process as the rest of the charges) whether or not he gives access. This differentiation of assemblies serves to avoid that in assemblies of private organizations are made votes of external people. For example, a public assembly may be that of a neighborhood association and a private assembly of a neighborhood community.

Selection of charges

In each assembly, as many charges are selected randomly as are required by the population density of the assembly and according to their reputation.

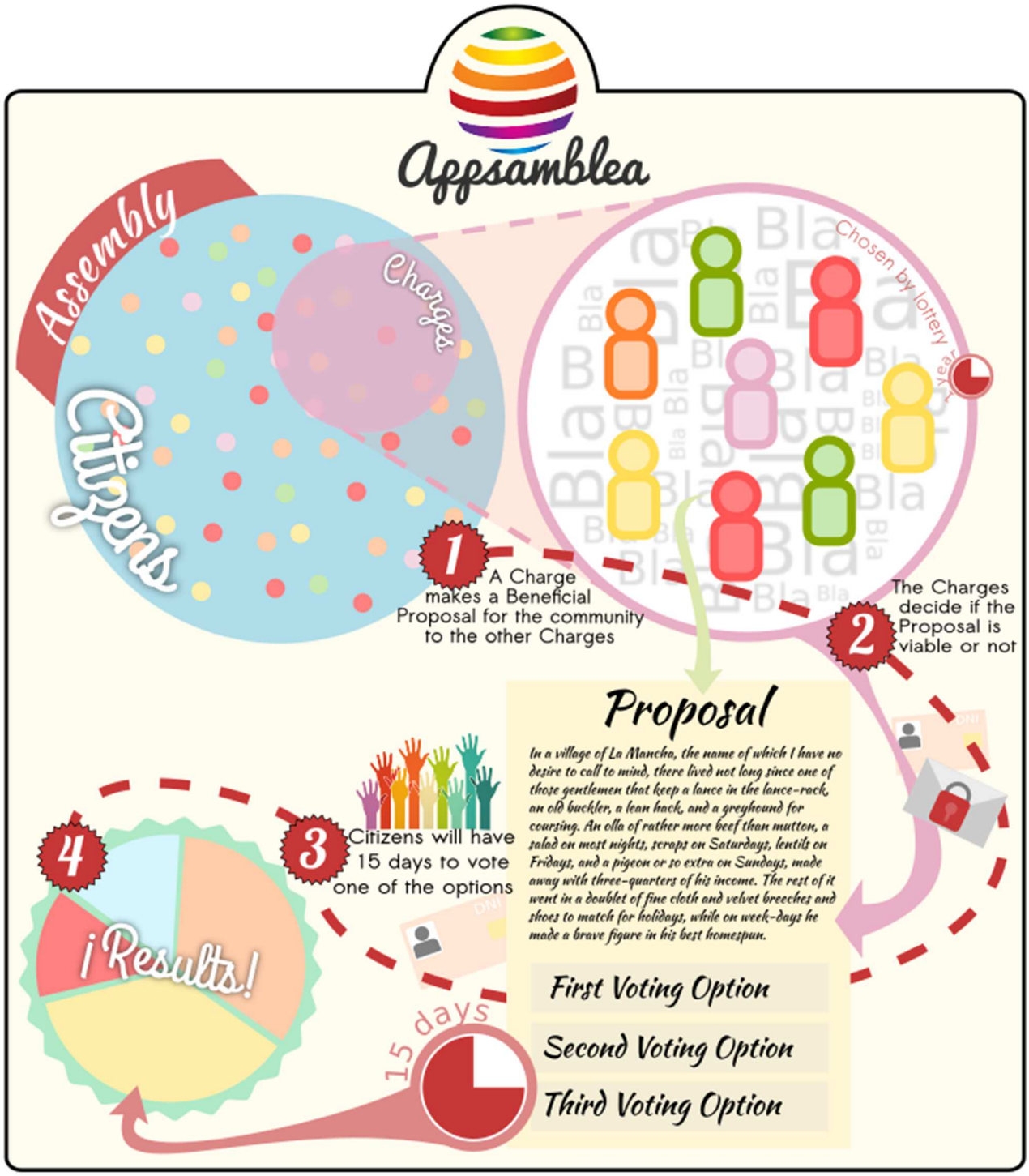
The randomness of the charges responds to the citizen representativeness of the same. According to the statesmen, to conduct a survey with a sample as representative as possible of a population, random extractions should be made to determine the people surveyed. This method of selection is the most efficient since the probability distributions associated with the draw depend only on the population on which it applies. For example, if the extraction variable were age we would have a Gaussian bell as a probability distribution. This is easily extrapolated to other variables that will have their own probability distributions. Moreover, when this type of selection is applied to a population, not only is a variable taken into account, but all possible ones within the population are selected. This is why, with the selection of charges through the draw, a sample of high representability is obtained [30].

Population density is the number of members that has an assembly. In the random selection of charges, a logarithmic function will be used to determine how many charges are required for each meeting. This function is used since it is inefficient to name a fixed percentage of charges in all the assemblies, this is because perhaps 25% is well in an assembly with few members, but in an assembly with about 10 thousand members would equal to have 2500 Charges, which would be a real organizational chaos.

The reputation system responds to the need to have positions prepared and aligned with the will of the members of the assembly. The reputation system will filter the eligible members of the ineligible, that is, a charge will be eligible when it exceeds the average reputation of all members of the assembly. The reputation will be won or lost according to factors that depend exclusively on the actions performed by a member within their virtual assembly: comments, votes received in comments, approval of proposals submitted or approved for voting, etc. This filtration through the gamification will allow a control of the quality of the charges and will encourage the participation in the assemblies.

Life cycle of a proposal

At the different stages of a proposal we will call it the "life cycle of a proposal". The next infographic collects these stages in a visual way, then the whole life cycle of a proposal will be explained in detail.



This infographic explains the voting process that is carried out in the application. The process consists of 4 steps in which a proposal goes through all its phases until it is ratified or repealed.

First, a charge formulates a proposal based on a need, solution or improvement that it deems appropriate for the community. This formulation involves the drafting of an explanatory and detailed text containing the entire development of the initiative. In addition, you must include the voting options that should be two or more. The voting options will depend on the type of proposal, for example, for a change of elevator in a neighborhood community, the proposal must contain at least two voting options with the budgets of the elevator companies.

The proposal, as a pre-proposal, is then ratified to vote for the other positions. That is, in order for the members of an assembly to vote on the voting options, before the other positions, must determine if it is a good proposal or not. Recall that the positions are chosen so that they are representative of the assembly. To overcome the pre-proposal phase must be ratified by at least half of the charges. This vote of ratification or repeal shall be made through the use of the DNIe so that the process is verifiable.

Once the proposal is accepted for the vote begins the period of fifteen days in which all members (including charges) will be able to vote using the electronic DNI the different voting options of the proposal. The deadline for voting on a proposal (D_n) is determined by the deadline of the last proposal (D_n -1):

- If D_{n-1} plus 3 days is less than the approval date plus 15 days, then D_n will be the approval date plus 15 days.
- Otherwise the deadline will respond to the equation: $D_n = D_{n-1} + 3$.

Using this control over the proposals prevents more than 5 simultaneous proposals from being accumulated in an assembly, which could cause user saturation that would have to separate their attention on too many issues.

When the voting period of fifteen days is completed, the polls are "closed" and counted. Being a fully computerized system the results are published instantly along with significant statistics of the voting.

Socioeconomic analysis

In this section, a distinction will be made between two key aspects of this FDP: system authentication through the use of DNIe and voting systems.

We will begin by analyzing the impact of the use of DNIe as a method of authentication in systems. At the moment the identification of users in systems is realized mainly by the use of a user name and a password. In addition to this method there are others such as the use of corporate identification cards, electronic keys, access tokens and other mechanisms. All these forms of authentication have an economic cost for any company, organization or administration as they have to use authentication servers, buy or create devices and cards for their members.

The Minister for Home Affairs, the General Directorate of the Police and the National Moneda and Timbre Factory created the DNIe project to give legal validity to the signing of the digital documents. Now, with the new DNIe 3.0, you can use all the functionality of this document identification by third parties, that is, it opens the development of applications and services. With this new approach the DNIe can be used as a tool for authentication in systems and thus make public and private entities save on infrastructure as well as improving their security.

Apart from the economic incentive to delegate the creation of identification elements and authentication systems, the use of DNIe as identification in systems implies that citizens will feel more inclined to use it given the facilities it entails. The fundamental point that would improve citizen security in the management of computer systems is that, through the use of DNIe users, in addition to having the document in their possession, they would only have to remember a password to authenticate themselves in web services. This would avoid typical notebooks or post-it with user credentials, which are one of the most serious dangers to the security of your information.

As for the voting systems, the current problems they have have been analyzed previously. The cost of validating a person's digital identity may be too great, for example, a system with a hundred thousand users that would have to validate them all would have no choice but to hire several people to carry out this process conscientiously, which would entail a brutal economic cost.

By using the DNIe these costs would cease to exist because using the electronic certificates ensures the identity of the users. The costs would be the ones that the public administration already has when creating the identification documents.

The social impact of electronic voting systems is evident in society, we are more and more and we are more and more eager to comment on the issues that surround us. It is through these platforms that citizens take part in these matters. It is clear that platforms for ideas and votes are something that is demanded from the sector of "smart-cities", which advocates that citizens are more involved in the development of their cities and communities. That is why a platform that uses the DNIe as a method of authentication would give legal validity to the votes cast by citizens. This legal validity is very necessary since the main criticism that is made to the existing platforms is that they can differentiate if a person has several accounts and emits more than one vote.

Conclusions and future work

Digital identity management I a major challenge for many public and private organizations that want their users to act unequivocally in their system. In Spain we have the advantage of a state agency that issues identification documents with digital certificates that can be used to solve this problem. The use of these documents, together with the current authentication systems such as OAuth2, is a clear need. Through the integration proposed by this FDP, it is possible to uniquely identify a person in connected systems.

It is visible that every day that passes the citizens we want to be more participants in the political life of our city and to be part of the decision making in our communities. That is why we are starting to use platforms for citizen participation in many of our cities. It is essential, therefore, to have open voting systems, which encourage the participation of all, without discrimination in its usability and that find synergies with

technologies of daily use. These platforms must have special security because they use sensitive data of an ideological nature. In addition, it is a priority that these platforms have voting verification systems to ensure that a person can only vote once.

The use of the DNIe at the corporate level is still stagnant due to the ignorance of these organizations of the great benefits that can grant them. Throughout the development of this FDP have seen different applications of third parties that already use this technology. My personal opinion is that it is a very powerful and resurgent technology, but it needs a small boost of adoption by technology companies.

Currently the development is only stable in Android with the DNIe v3.0, although it is working in a version of web client that has the same functionality with the same security and that serves for any version of the electronic DNI. This step is necessary since the technological restrictions of the current version greatly limit its use by the users.

In addition, to avoid data centralization, the entire server side will be migrated to a Blockchain distributed environment in the near future. This environment will improve the security of data and improve the auditing of votes by external entities. As Blockchain allows its nodes to have no confidence between them, anyone interested in improving the capabilities of the system using their systems, it will be possible to increase power without increasing costs.

To conclude, I have to analyze the personal benefits that the development of this work has brought me. First, and as I said at the beginning, I had this project in mind for a long time, but because of personal reasons, time and technology (there was still no NFC implementation for Android) I could not develop it. It is with this FDP with which I have been able to begin to outline a long-distance project and of which I hope that more people will join after realizing this little proof of concept. Teaching the final result of my work, I have realized that people demand this kind of initiatives and for this reason I will continue to develop this project in the future to be as useful as possible to all those citizens who want to participate and improve their community.

Índice

Índice de tablas

Índice de ilustraciones

1. Introducción

La elección mediante voto es la base de un sistema democrático y, por lo tanto, clave en nuestra sociedad. La tecnología, que ha avanzado mucho en otros temas, sin embargo, aún no ha logrado ofrecer una alternativa viable a este tipo de sistemas tradicionales.

> [RAE] Voto:
>
> 2. m. "Gesto, papeleta u otro objeto con que se expresa una preferencia ante una opción"

A lo largo de la historia de las Tecnologías de la Información han sido varios los que han propuesto soluciones para implementar sistemas de voto. Uno de los problemas que se encuentran es la variedad de modelos de votación que hacen que exista poca sinergia entre ellos (y por tanto sean más costosos). Aun así, hay ciertos aspectos de todos estos sistemas que presentan puntos comunes, como, por ejemplo, los sistemas de autenticación del votante y la posterior validación de votos. Son estos aspectos en los que se centrará este Trabajo de Fin de Grado.

Los requisitos principales de estos sistemas son que deben ser muy seguros, accesibles para todos, auditables y muy relacionado con esto último, de código abierto. Seguros porque los datos que se manejan son de carácter personal y protegidos por la Ley Orgánica de Protección de Datos [1] como datos muy sensibles al contener información sobre afiliaciones políticas. Han de ser accesibles de manera que cualquier ciudadano puedan usarlas independientemente de su condición física y edad. Auditables por entidades externas e independientes que verifiquen el resultado de una elección de manera clara y que no dé lugar a dudas de la veracidad del voto. Si son de código abierto, además permite a los auditores saber lo que hace el sistema y proporcionar una transparencia fundamental.

Existe un caso de éxito de voto electrónico el cual se considera referente mundial, este es el caso de Estonia donde los ciudadanos pueden votar por Internet mediante un carnet de identidad parecido al español.

> [RAE] Identidad:
>
> 2.f. Conjunto de rasgos propios de un individuo o de una colectividad que los caracterizan

La identidad digital es el rastro que deja un usuario mediante la interacción con servicios digitales. Como usuarios de Internet, nuestro rastro digital es bastante evidente cada vez que realizamos registros en webs, subimos contenido personal, expresamos nuestra opinión, etc. Esto es fácilmente comprobable con una simple búsqueda de nuestro nombre en cualquier buscador [2], [3].

El problema viene a la hora de verificar si esa información es válida y proporcionada por el usuario que utiliza estos servicios digitales. La solución que la mayoría de empresas ha tomado es basarse en la información que otros proveedores ofrecen, véase Facebook o Google que permiten la autenticación de usuarios mediante sus servicios evitando que los usuarios tengan que crear nuevas identidades en cada servicio que quieran utilizar.

Estos sistemas tienen dos problemas importantes: el primero es que no proveen de un mecanismo que verifique que una identidad personal es única, es decir, una persona puede tener varias cuentas en uno de estos servicios; el segundo es que estos proveedores no ofrecen un servicio unificado y por tanto terminan por crearse varias identidades dependiendo del proveedor que el usuario elija en el momento de autenticarse en otra web.

Existen, como no, alternativas viables, pero poco conocidas. “OpenID es un estándar de identificación digital en el que un usuario puede identificarse en una página web a través de una URL y puede ser verificado por cualquier servidor que soporte el protocolo”[4]. Esto resuelve el problema de los servicios unificados, pero no el de la identidad única. Además, ha recibido bastantes críticas con respecto a la seguridad y privacidad relativas a la confidencialidad de los servicios consumidos y que ante el compromiso de la seguridad del dispositivo del usuario toda su información podría quedar desprotegida.

1.1. Motivación

Existe mucha controversia en si es posible realizar un sistema de votación en España con los medios a nuestro alcance [5]. La motivación que impulsa este Trabajo de Fin de Grado es tener un primer contacto con las tecnologías que harían posible la creación de estos sistemas conforme a la regulación vigente en España y determinar su viabilidad a la hora de implementarlos.

Como motivación personal, he de añadir que es un proyecto que tenía en mente desde hace bastante tiempo y este Trabajo de Fin de Grado me ofrece la posibilidad de dedicarle un tiempo y esfuerzo que en otra situación no podría dedicarle.

1.2. Objetivos

El principal objetivo de este Trabajo de Fin de Grado (TFG) es analizar, diseñar e implementar un sistema de voto cuya autenticación de votantes y validación de votos se realice mediante el uso de los certificados incluidos en el Documento Nacional de Identidad.

Como objetivo adicional, trataré de estudiar distintos mecanismos de voto electrónico y probar distintas soluciones comerciales que ofrecen este servicio. Me centraré, sobretodo, en como solucionan el problema de la identidad digital y la verificabilidad de voto que ofrecen.

1.3. Estructura del documento

Este documento tratará de explicar todo lo referente a este TFG de acuerdo a la siguiente estructura de apartados generales:

- **Introducción**: Esta es la sección con la que comienza el documento y en la que se hace una presentación del problema a resolver, se establecen los objetivos y las motivaciones; y se explica la estructura del documento.
- **Estado del arte**: En este apartado se presentarán y analizarán las distintas tecnologías que afectan a este TFG. Se prestará especial atención al voto por Internet, el Documento Nacional de Identidad, la tecnología NFC y la presentación de algunas plataformas de voto existentes.
- **Análisis**: A lo largo de este apartado se analizarán las expectativas del sistema que se va a desarrollar. Se hará una determinación del alcance del sistema, una especificación de estándares y normas; y, por último, el establecimiento de requisitos de software
- **Diseño**: Este apartado recogerá el funcionamiento del sistema en base al análisis previo. Se diseñarán las partes implicadas en el sistema independientemente de su posterior implementación.
- **Implementación y pruebas**: Esta sección realizará una exhaustiva descripción de cómo se ha llevado a cabo la implementación del problema planteado. Después, se definirán un conjunto de casos de prueba necesarias para satisfacer el correcto funcionamiento.
- **Metodologías usadas**: Este apartado recoge las distintas metodologías que se han utilizado a lo largo del desarrollo del proyecto.
- **Marco regulador y aspectos legales**: Sección que recoge y analiza las distintas regulaciones legislativas que pueden ser aplicables al proyecto.
- **Análisis socioeconómico**: Apartado que analiza el impacto social y económico del proyecto.
- **Presupuesto y planificación**: Apartado que desarrolla la planificación realizada a lo largo de todo el proyecto. El presupuesto recogerá el coste de realización e implantación del proyecto.
- **Conclusiones y trabajo futuro**: En este apartado se presentan las conclusiones sobre el proyecto y se expondrán mejoras y expansiones sobre el mismo.
- **Glosario**: Sección con definiciones y abreviaturas.
- **Referencias**: Apartado con las fuentes de información utilizadas.

2. Estado del arte

El voto electrónico puede definirse como el proceso que permite a un individuo ejercer su voto en un medio electrónico. Surge como herramienta para mejorar y facilitar el acceso al voto mediante el uso de las tecnologías de la información. En la actualidad, existen múltiples métodos de votación, ya sean tarjetas perforadas, votación con escáner óptico, votación con certificados, etc [6]. Este Trabajo de Fin de Grado se centrará exclusivamente en los sistemas de voto por Internet.

Los sistemas de voto por Internet proporcionan muchas ventajas tanto a las instituciones organizadoras como a los votantes. Para las instituciones, conlleva principalmente un ahorro importante en el gasto que implica movilizar recursos tanto materiales como económicos a la hora de realizar cualquier tipo de votación. Para los votantes, supone una mejora ya que no han de desplazarse hasta una urna tradicional satisfacer su derecho a voto [7]. Sobre las ventajas y desventajas de estos sistemas de voto hablaremos detalladamente más adelante.

Muchos de estos sistemas de voto utilizan *Smart-cards* para realizar la verificación de identidad de los votantes. En España, contamos con el Documento Nacional de Identidad Electrónico (DNIe) pensado para adaptar el antiguo DNI a la sociedad de la información y el consumo de servicios electrónicos. Comenzó a expedirse en 2006 y cuenta con un chip criptográfico que proporciona certificados de autenticación y firma. Este chip posee medidas de seguridad digitales como el cifrado de datos del chip y el acceso a la funcionalidad interna mediante una clave conocida únicamente por el ciudadano. Además, cabe destacar que las claves nunca abandonan el chip. Más adelante se detallarán los aspectos técnicos de este documento.

2.1. El voto por Internet

2.1.1. Tipos de voto electrónico

Antes de comenzar a hablar exclusivamente del voto por Internet, que es el objeto de este TFG, comenzará por explicar brevemente que tipos de voto electrónico existen y en qué consisten.

Hay dos grandes grupos de voto electrónico, el voto físico y el voto remoto.

El voto físico se realiza presencialmente en un centro de votación, en España concretamente en los colegios electorales. Este tipo de votación conlleva la supervisión por las autoridades electorales que evitan que ocurran coacciones, fraudes y otras incidencias. Este tipo de voto electrónico se lleva a la práctica fundamentalmente con máquinas de votación especiales donde el votante elige en una pantalla la opción de su elección.

El voto remoto implica que el elector puede votar a distancia y sin presencia de supervisión. Además, ha de adaptarse a los dispositivos y mecanismos que los votantes tengan a su disposición. En todo caso este tipo de votación debería siempre ir acompañada del modelo de voto físico [7], [8].

2.1.2. Historia del voto por Internet [9]

La historia sobre el voto por Internet se remonta a principios 2000 cuando se empiezan a realizar pruebas por parte del *Federal Voting Assistance Program (FVAP)* perteneciente al Departamento de Defensa de los Estados Unidos (DD) ya que el voto a distancia se estaba realizando hasta entonces mediante fax. Este programa consiguió dar forma a una prueba de concepto de un sistema de voto por Internet que fue usado en unas elecciones reales, lo cual, sentó no solo una base teórica sino también práctica.

Esta prueba de concepto abrió el camino a más elecciones por Internet. Prueba de ello fueron las elecciones generales de 2004 en Estados Unidos donde cincuenta y cinco condados se presentaron voluntarios para participar en *Secure Electronic Registration and Voting Experiment (SERVE).*

A pesar del entusiasmo del equipo responsable, el proyecto no contaba con el apoyo de *Security Peer Review Group (SPRG)* el cual se mostraba especialmente crítico con estos sistemas de voto. Así pues, a finales de 2004 se extendió el mito de la inseguridad del voto sobre Internet que ha prevalecido hasta no hace mucho.

El resurgir del voto por Internet se dio como consecuencia del caso de éxito de Estonia en 2007. Estonia se convirtió en la primera nación en desarrollar legislación para unas elecciones generales mediante el voto por Internet, tras el éxito del proyecto piloto en las elecciones municipales de 2005. A partir de entonces otros países y asociaciones han ido adoptando, con moderación, estos sistemas de voto.

Existen en la actualidad numerosas plataformas privadas que intentan desarrollar sistemas que implementen el voto electrónico para dar soporte a elecciones dentro de organismos o empresas. Se desarrollarán más adelante en este mismo documento.

2.1.3. Ventajas e inconvenientes del voto por Internet

En este apartado se tratará de resumir y analizar las distintas ventajas e inconvenientes que presentan estos sistemas [10], [11].

Ventajas

- Coste. El coste de unas elecciones por Internet bajaría enormemente.
- Permite el voto desde cualquier lugar que tenga acceso a Internet.
- Permite el voto a personas con discapacidad o dificultad de movilidad al colegio electoral.

- Mejora del voto por correo que requiere muchos recursos e infraestructura.
- Impacto medioambiental. Se evitaría la utilización de recursos materiales que serían innecesarios como, por ejemplo, el papel de las papeletas y listas.
- Estadísticas en tiempo real. Desde el inicio serían accesibles la participación y otros datos no relativos al recuento.
- Velocidad de escrutinio. Sería posible conocer el resultado de la votación nada más acabar el tiempo disponible.
- Fomenta la participación ciudadana.

Inconvenientes

- La seguridad en estos sistemas es la principal desventaja ya que al depender de sistemas informáticos expuestos a Internet pueden ser objetivo de ataques.
    - Seguridad en auditoría de voto.
    - Posible fraude por parte de intrusos.
- Acceso al voto por parte de usuarios no acostumbrados al uso de las tecnologías de la información o incapacitados para hacerlo.
- Coacción al voto.

Como podemos observar, las ventajas que ofrece el voto por Internet son más que las desventajas. Aun así, hay que prestar atención a estos posibles inconvenientes ya que afectan directamente a la seguridad, privacidad y usabilidad de estas tecnologías.

La seguridad debe ser primordial en este tipo de sistemas, ya que afecta a datos sobre afiliación política que son sensibles en gran medida. Es por ello que es la mayor crítica que se le hace al voto por Internet. Aun así, ya existen mecanismos y tecnologías que proporcionan esta seguridad.

La seguridad durante la auditoría de voto ha de ser robusta, es decir, la autoridad encargada del recuento de votos no debe tener acceso a la identidad del votante a la vez que el votante sea capaz de ver su voto. Existen multitud de soluciones que dan respuesta a este problema, desde sistemas centralizados con políticas muy cerradas de acceso a la información a sistemas descentralizados y de software libre como smart-contracts en redes Blockchain.

Ante posibles fraudes por intrusiones del sistema, es decir, ataques de ciberdelincuentes con ánimo de modificar los resultados de votaciones, también existen varias soluciones. Éstas van desde el clásico control

de las redes, permisos del sistema hasta las anteriormente mencionadas redes Blockchain que no necesitan confianza entre sus nodos. Todo esto, unido a una buena planificación e ingeniería de la seguridad desde el inicio de los desarrollos, hace que estos ataques sean poco probables y en caso de darse que los daños causados sean mínimos y rectificables.

Los argumentos sobre la inseguridad de estos sistemas están bien fundados, pero mediante un correcto diseño, ingeniería, desarrollo y protocolos de seguridad se puede paliar, e incluso erradicar estos vectores de ataque.

En cuanto a la accesibilidad a los mecanismos de voto por Internet, existen mecanismos como el uso de cajeros automáticos o de voluntarios que ya sea mediante llamada telefónica o visita en persona permiten votar a las personas mayores, discapacitados y cualquier persona que no pueda o sepa usar las nuevas tecnologías.

La comparación entre ventajas e inconvenientes nos deja patente que las ventajas que ofrecen estos sistemas benefician claramente a nuestras sociedades democráticas y que los inconvenientes que conllevan son sobretodo de carácter técnico y tienen solución.

2.2. Documento Nacional de Identidad Electrónico (DNIe)

2.2.1. Historia del DNI [12]

La historia del Documento Nacional de Identidad (DNI) nace de la necesidad del Estado Español del militar y dictador Francisco Franco Bahamonde de tener más y mejor controlados a los españoles. No fue hasta 1951 cuando se empezaron a expedir los primeros documentos identificativos comenzando por los presos, ciudadanos en libertad vigilada o ciudadanos que, por motivos laborales, habían de desplazarse con asiduidad. A partir de entonces se empezó a expedir en las ciudades con más población extendiéndose más tarde a poblaciones más pequeñas.

Al igual que hoy en día, los números del DNI eran asignados a equipos de expedición, los cuales se encargaban de verificar la información que el ciudadano otorgaba. Al contrario de lo que la creencia popular afirma, los números bajos no pertenecen a fallecidos, sino que son parte del lote correspondiente al equipo de expedición donde se obtuvo el documento.

A lo largo de su historia ha cambiado hasta siete veces de diseño y características. Desde el primero, que lo conformaba una cartilla en papel grueso que contenía los datos de filiación, profesión y categoría económica además de los datos habituales como nombre y apellidos, hasta el actual que cuenta con chip criptográfico y tecnología NFC.

Este documento hace partícipe al ciudadano titular en la obligación de su custodia y conservación ya que el DNI es legalmente acreditativo de la identidad de dicha persona.

2.2.2. Características electrónicas [13]

- Chip
    - Modelo del Chip:SLE78CLFX408AP de Infineon Technologies
    - Sistema operativo: DNIe v4.0 /Versión comercial DNI 3.0)
    - 400 KB memoria Flash (código + personalización)
    - 8 KB memoria RAM
    - Dual Interface (con contacto / sin contacto).
    - Criptolibrería RSA
    - CC EAL5+
- Contenido
    - ZONA PÚBLICA: Accesible en lectura sin restricciones, contenido:
        - Certificado CA intermedia emisora.
        - Claves Diffie-Hellman.
        - Certificado x509 de componente.
    - ZONA PRIVADA: Accesible en lectura por el ciudadano, mediante la utilización de la Clave Personal de Acceso o PIN, conteniendo:
        - Certificado de Firma (No Repudio).
        - Certificado de Autenticación (Digital Signature).
    - ZONA DE SEGURIDAD: Accesible en lectura por el ciudadano, en los Puntos de Actualización del DNIe.
        - Datos de filiación del ciudadano (los mismos que están en el soporte físico).
        - Imagen de la fotografía.
        - Imagen de la firma manuscrita
    - DATOS CRIPTOGRÁFICOS: Claves de ciudadano
        - Clave RSA pública de autenticación (Digital Signature).
        - Clave RSA pública de no repudio(ContentCommitment).

- Clave RSA privada de autenticación (Digital Signature).
- Clave RSA privada de firma (ContentCommitment).
- Patrón de impresión dactilar.
- Clave Pública de root CA para certificados card-verificables.
- Claves Diffie-Hellman.

- DATOS DE GESTIÓN:
  - Traza de fabricación.
  - Número de serie del soporte.

2.2.3. Firma electrónica con DNIe

Según la Ley 59/2003 del 19 de diciembre de 2003, el Documento Nacional de Identidad iguala la validez jurídica de la firma electrónica a la firma tradicional [14].

> Ley 59/2003, de 19 de diciembre
> Artículo 3, punto 4: "La firma electrónica reconocida tendrá respecto de los datos consignados en forma electrónica el mismo valor que la firma manuscrita en relación con los consignados en papel."

Además de identificarnos de manera presencial, también nos puede identificar de forma telemática haciendo posible su uso en la identificación en páginas web gubernamentales, servicios de banca, autenticación de usuarios en equipos, firma electrónica de documentos, etc.

Lo único que se necesita para poder firmar con el DNIe v3.0 es:

- **PC**:
  - Lector de tarjetas digitales y sus drivers.
  - Controladores del DNIe (Descargables en la página oficial del DNIe).
  - DNIe y contraseña de los certificados.
- **Android**
  - Teléfono con tecnología NFC.

- Aplicación de firma electrónica.
- DNIe y contraseña de los certificados.

El nuevo DNIe v3.0, al incluir la tecnología NFC, permite la lectura y firma inalámbrica con los certificados digitales y hará posible infinidad de nuevos desarrollos (sobretodo móviles) que permitan facilitar procesos legales, organizativos y administrativos.

En este punto, les toca mover ficha a las empresas que deberán adoptar el uso de esta tecnología en la firma de documentación y sistemas de autenticación más robustos[15].

2.3. Near Field Communication (NFC)

Near Field Communication es una tecnología wireless derivada de las etiquetas RFID (Radio Frecuency Identification) que trabaja en la banda de los 13,56 Mhz pensada desde su inicio para su uso en smartphone. Tiene una tasa de transferencia de 424 kbit/s, por lo que no tiene gran utilidad en la transmisión de gran cantidad de datos, aun así, es muy útil en los casos de uso de identificación y validación de equipos y personas [16], [17].

Esta tecnología tiene principalmente dos modos:

- **Activo**: en el cual dos dispositivos comparten datos mediante la emisión de una señal. Funciona de forma similar al Bluetooth.
- **Pasivo**: en el cual solo hay un dispositivo emitiendo y el otro aprovecha esta emisión para el intercambio de datos

Al ser una tecnología de transmisión, la seguridad de los datos radica sobretodo en la cercanía necesaria entre los dos dispositivos. Es posible proteger los datos mucho mejor si las transacciones utilizan SSL, pero esto no es intrínseco a NFC, sino que tendrán que implementarlo los servicios que utilicen esta tecnología.

Los principales casos de uso para los que ya es utilizado son [18]:

- **Identificación**: Mediante una tarjeta (pasivo) o el Smartphone (activo) se puede controlar el acceso a instalaciones.
- **Lectura de datos**: Mediante la lectura de elementos pasivos se pude obtener información de un producto o incluso participar en juegos de orientación.

- **Pago con smartphone**: Pagar usando el móvil ya es una realidad gracias a esta tecnología. Basta con acercar el smartphone al TPV. Es el caso de uso con más proyección ya que tiene detrás a las entidades bancarias.

En este TFG se hace uso del caso de uso de identificación utilizando el chip NFC que tiene el DNIe v3.0. Este chip, extrae los certificados criptográficos y permite realizar las operaciones de firmas necesarias para la autenticación. La seguridad que ofrece en la conexión es de clave simétrica donde, para establecer comunicación, es necesario conocer el número CAN impreso en la parte delantera de la tarjeta, esto dará acceso tan solo a los datos públicos. Para la obtención de los certificados digitales y el acceso a la funcionalidad de firma, además, es necesario conocer la contraseña del DNIe.

Existe cierta teoría de la conspiratoria con respecto a que con el nuevo DNIe la policía puede controlar masas desde la distancia. En primer lugar, para obtener acceso al DNIe es necesario estar a corta distancia, además es necesario poseer como mínimo el número CAN para acceder a la información pública del documento. Por tanto, estas teorías no tienen ningún fundamento técnico.

2.4. Plataformas existentes

2.4.1. Ideascale

Ideascale es una plataforma online que permite a las organizaciones involucrarse en las opiniones que tienen las comunidades mediante la recolección de ideas y la votación de los usuarios de las mismas. Es una empresa que ofrece crowdsourcing tanto a empresas públicas como privadas [19].

En 2009, Vivek Bhaskaran y Rob Hoehn aprovecharon la Iniciativa de Gobierno Abierto del Presidente Obama para lanzar Ideascale. Tan solo en el primer año fue adoptada por 23 agencias federales y ha sido utilizada por muchas organizaciones. Los años posteriores consiguieron aumentar en gran medida tanto organizaciones públicas como privadas gracias a su facilidad de uso y bajo coste[20].

Para poder registrarse en el sistema es necesario tener una cuenta de correo de una organización. Utilizando la cuenta de correo de la universidad se puede acceder a la comunidad de la UC3M de Ideascale y utilizarla para proponer y votar ideas. Las identidades personales que se generan están directamente relacionadas con el correo de la organización, es decir, si se tienen varias cuentas de correo en la organización se podría votar varias veces. De esta manera Ideascale se “lava las manos” en cuanto a la validez de la identidad de los usuarios.

Cualquier miembro de una comunidad puede proponer ideas. En Ideascale promueven, mediante la gamificación, que sus usuarios sean partícipes de la comunidad. Además de plantear ideas y votarlas, esta plataforma ofrece foros de discusión donde cualquiera puede opinar.

En cuanto al sistema de votación, es tan sencillo como clicar en un botón de "+1 voto" o "-1 voto" sin ningún tipo de validación. Esto tiene una ventaja en la facilidad de uso de la aplicación ya que es muy sencillo votar, pero implica un problema muy grande de autenticación del voto, es decir, no hay manera de saber si realmente el que realiza la votación es el usuario o es otra persona aprovechando una sesión indebidamente abierta.

Ideascale proporciona una interfaz de usuario muy amigable e inteligible que organiza las ideas entorno a campañas y las filtra por distintos factores como: últimas, popular, candente, mis ideas y aleatorio.

2.4.2. nVotes (Agoravoting)

NVotes es la nueva marca de Agoravoting [21]. Esta plataforma ofrece servicios de votaciones sobre Internet a cualquier organización de forma segura, flexible y transparente. Algunos partidos políticos españoles lo utilizan para realizar consultas internas en las que los militantes puedan opinar sobre cuestiones organizativas [22].

Agoravoting se creó en 2014 por unos integrantes del Partido de Internet. En una entrevista, uno de sus creadores afirmó que las primarias que convocó el partido Podemos antes de las elecciones europeas de 2014 fue el primer negocio de la compañía. Después, otros partidos han utilizado la plataforma de forma interna e incluso para definir cursos de acción a nivel nacional (Joan Valdoví) [23].

Su modelo de negocio se basa en facturar en función de la seguridad y el proceso de autenticación y verificación que requiera la implantación de la votación. El precio suele oscilar entre los 40 y 50 céntimos por votante.

Los procesos de autenticación y verificación que utilizan son:

- **Escaneado del DNI**: Se verifica la identidad del votante mediante una imagen escaneada del Documento Nacional de Identidad.
- **Contraseña por correo**: Se envía una contraseña única a un correo del votante que servirá para una única votación.
- **Contraseña por SMS**: Este método es igual que el anterior pero la contraseña se envía mediante SMS a un número de teléfono del votante.

La funcionalidad más destacada es que Agoravoting cifra el voto en el cliente web mediante Javascript y no se descifra hasta que la votación se cierra. También verifica matemáticamente el recuento y mantiene la privacidad del votante [24].

Esta plataforma es la más importante en España y un referente mundial en cuanto a votaciones electrónicas de cierta seriedad. Se utiliza tanto en organizaciones como en partidos políticos y por ello es necesario que tenga una seguridad de primer nivel. Aun así, falla estrepitosamente en los mecanismos de autenticación y verificación, ya que existen múltiples métodos de sortear las medidas que ofrecen y bien falsificar la información o votar varias veces (tener varias cuentas de correo o números de teléfono).

2.4.3. Change.org

Change.org es una organización que vende y presta servicios mediante envíos masivos, listas de email y servicios parecidos. Actúa como un blog y permite publicar peticiones online.

Esta plataforma se lanzó en febrero de 2007 siendo un referente en todo el mundo haciendo efectivas muchas de las peticiones. Desde entonces, se ha convertido en la mayor plataforma de peticiones, logrando que incluso organismos oficiales los utilicen para conocer la opinión de sus ciudadanos[25].

Su modelo de negocio se basa en vender los datos de las personas que votan, números de teléfono, email y otras estadísticas por las que otras empresas están interesadas. Por ejemplo, si una empresa decide crear una campaña, pagará una cierta cantidad a Change.org y la plataforma la difundirá y reportará los datos de los que firmen. Estos datos son muy importantes para organizaciones no gubernamentales que se financian a través de donaciones, puesto que según Change.org la gente que firma es más proclive a donar[26].

Una de las críticas a esta compañía es que no publican sus cuentas a pesar de afirmar hacer una labor social y transparente. Otra crítica a Change.org es que aloja multitud de peticiones en contra de la evasión fiscal mientras que tiene su sede en Delaware, donde la política fiscal es muy laxa.

Esta plataforma, aunque es de las más grandes en cuanto a usuarios y poder de influencia, tiene un gran punto débil, las firmas a las peticiones no son legalmente válidas. No son legalmente válidas porque no hacen validación alguna de los firmantes, es decir, cualquiera puede votar cualquier petición e incluso hacerlo varias veces si se poseen varias cuentas.

2.4.4. Consul

Consul es la plataforma desarrollada originalmente para el programa de participación ciudadana de Decide Madrid. Esta iniciativa permite a los ciudadanos debatir, proponer y decidir sobre temas de su ciudad. A través de este sistema, los ayuntamientos pueden hacer un gobierno más abierto a la comunidad y que permita participar activamente en la política de sus ciudades a todos los interesados [27], [28].

A parte de Madrid, que fue pionera en adoptar ese sistema, actualmente hay 37 ayuntamientos, diputaciones y universidades, tanto nacionales como extranjeras utilizando esta plataforma.

El sistema informático sobre el que se sustenta esta herramienta de participación, es software libre con licencia AGPL y de libre implantación. Es decir, las organizaciones que quieran utilizarlo podrán modificarlo y adaptarlo a sus necesidades sin pagar nada. Además, es una plataforma configurable en estilo y diseño que se adapta fácilmente a las necesidades. Se puede contactar con los creadores para pedir ayuda para ponerlo en marcha [29].

La plataforma cuenta con varias funcionalidades entre las cuales se encuentran los debates, que son como foros especializados por tema; las propuestas que necesitan pasar un filtro de apoyos para parar a votación; una sección de presupuestos participativos en los que los ciudadanos pueden destinar partidas presupuestarias a proyectos concretos; y ,finalmente, las votaciones a las propuestas que usan una verificación de usuarios evitando la duplicidad de votos También ofrecen la posibilidad de restringir la participación por distritos / barrios. La plataforma tiene más funcionalidades como la legislación colaborativa y los procesos sectoriales que permiten decidir a los ciudadanos sobre asuntos más complejos.

La principal crítica que se le hace a esta plataforma es la verificación de los votantes, ya que para votar por Internet lo único necesario es conocer el número de DNI y el código postal de la persona por la que un intruso quiera hacerse pasar.

El punto fuerte de esta herramienta es que permite la inclusión de los resultados de votaciones en urnas, esquivando en gran medida las dificultades de ciertos sectores de la ciudadanía que no tienen o pueden acceder al servicio telemático.

3. Análisis

En este punto se ofrece una descripción de los problemas que resuelve el sistema, con qué sistemas interactúa y quienes son los usuarios que van a usarlo.

3.1. Determinación del alcance del sistema

Podemos definir Appsamblea como un sistema de voto electrónico. Su objetivo será facilitar la recolección, recuento y visionado de votaciones en una tipología de democracia directa orientada a todo tipo de agrupaciones sociales. Concretamente, los usuarios podrán crear una cuenta en la aplicación con su Documento Nacional de Identidad (DNI), crear asambleas con las que organizarse con otros usuarios y votar propuestas usando la firma electrónica con el DNIe. La funcionalidad completa se definirá a lo largo de este apartado de análisis.

3.1.1. Asambleas

El sistema organiza a los ciudadanos en asambleas virtuales dentro de su zona de actuación. La zona de actuación de un ciudadano corresponde a la información del censo incluida en su DNI, es decir, si un ciudadano es de Toledo sus zonas de actuación serán Toledo, la provincia de Toledo, la Comunidad Autónoma de Castilla La-Mancha y España.

Podemos definir asamblea virtual como el lugar no físico donde se reúnen las personas para opinar sobre asuntos comunes. Una asamblea puede estar constituida en cualquier organización, empresa, comunidad de vecinos, equipo de fútbol, etc.

Además, las asambleas podrán ser públicas o privadas. Las primeras serán accesibles para todo el mundo, cualquiera que quiera ser miembro podrá serlo. Las segundas, en cambio, no son accesibles a todo el mundo, y dependerá del administrador (el cual es sometido al mismo proceso de selección que el resto de cargos) si le da acceso o no. Esta diferenciación de asambleas sirve para evitar que en asambleas de organizaciones privadas se realicen votaciones de gente externa. Por ejemplo, una asamblea pública puede ser la de una asociación vecinal de barrio y una asamblea privada la de una comunidad de vecinos.

3.1.2. Selección de cargos

En cada asamblea se seleccionan aleatoriamente tantos cargos como sean requeridos por la densidad poblacional de la asamblea y en función de su reputación.

La aleatoriedad de los cargos responde a la representatividad ciudadana de los mismos. Según los estadistas, para realizar una encuesta con una muestra lo más representativa posible de una población, se deben realizar extracciones aleatorias para determinar la gente encuestada. Este método de selección es el más eficiente ya que las distribuciones de probabilidad asociadas al sorteo dependen únicamente de la población sobre las que aplica. Por ejemplo, si la variable de extracción fuera la edad tendríamos una

campana de Gauss como distribución de probabilidad. Esto es fácilmente extrapolable a otras variables que tendrán sus propias distribuciones de probabilidad. Es más, cuando este tipo de selección se aplica a una población no solo se toma en cuenta una variable, sino que se seleccionan todas las posibles dentro de la población. Es por ello que, con la selección de cargos mediante el sorteo, se obtiene una muestra de alta representatividad [30].

La densidad poblacional es la cantidad de miembros que tiene una asamblea. En la selección aleatoria de cargos, se utilizará una función logarítmica para determinar cuántos cargos son necesarios para cada asamblea. Se utiliza esta función ya que es poco eficiente nombrar un porcentaje fijo de cargos en todas las asambleas, esto es así porque quizás el 25% está bien en una asamblea con pocos miembros, pero en una asamblea con unos 10 mil miembros equivaldría a tener 2500 cargos, lo que sería un verdadero caos organizativo.

El sistema de reputación responde a la necesidad de tener cargos preparados y alineados con la voluntad de los miembros de la asamblea. El sistema de reputación se encargará de filtrar a los miembros elegibles de los no elegibles, es decir, un cargo será elegible cuando supere la media de reputación de todos los miembros de la asamblea. La reputación se ganará o se perderá de acuerdo a unos factores que dependan exclusivamente de las acciones que realiza un miembro dentro de su asamblea virtual: comentarios, votos recibidos en comentarios, aprobación de propuestas presentadas o aprobadas para su votación, etc. Esta medida de filtrado mediante la gamificación permitirá un control de la calidad de los cargos y fomentará la participación en las asambleas.

3.1.3. Ciclo de vida de una propuesta

A las diferentes etapas de una propuesta las llamaremos “ciclo de vida de una propuesta”. La siguiente infografía recoge estas etapas de una manera visual, después se explicará detalladamente todo el ciclo de vida de una propuesta.

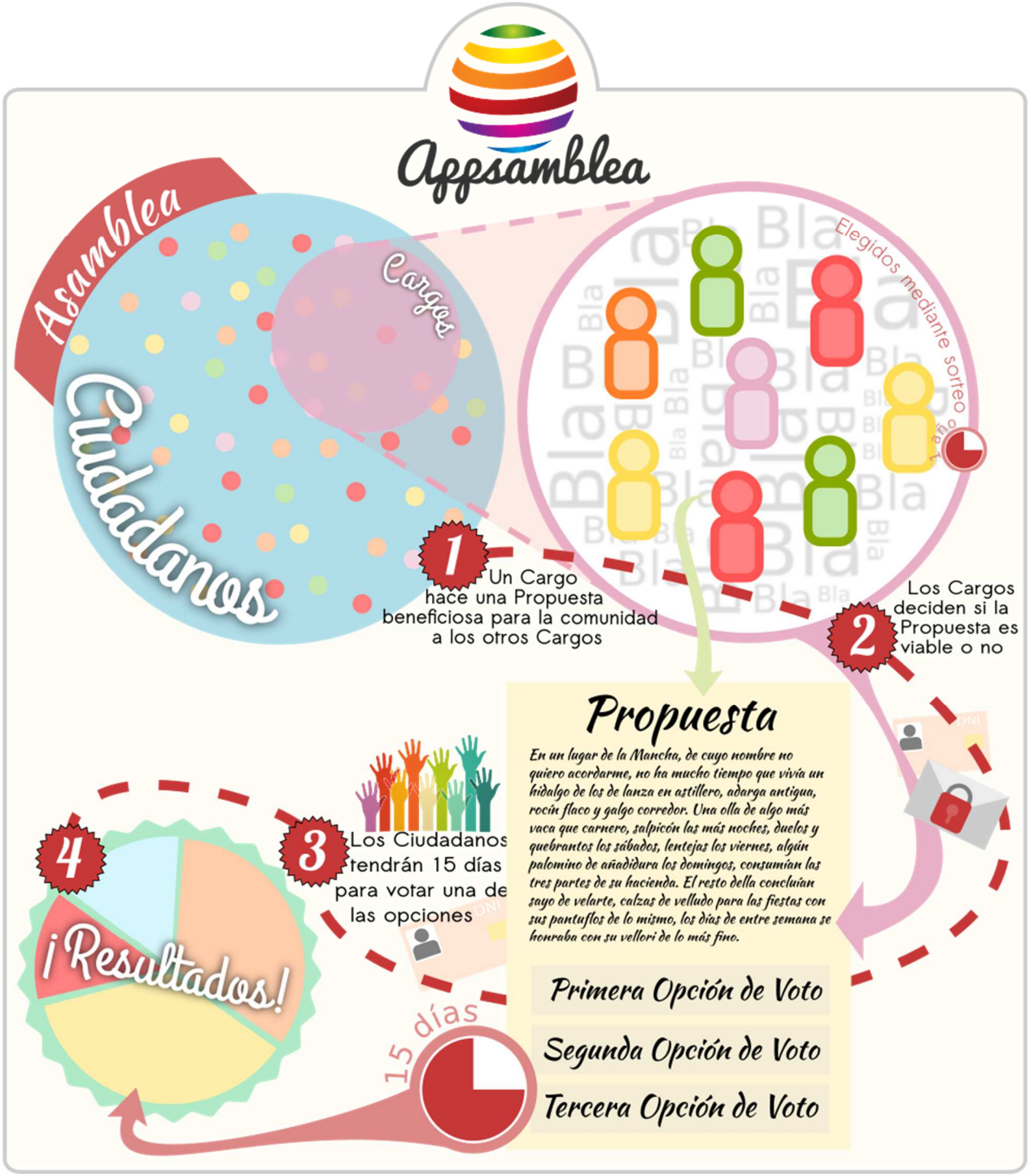


Ilustración 1: Infografía del ciclo de vida de una propuesta

Esta infografía explica el proceso de voto que se realiza en la aplicación. El proceso consta de 4 pasos en los cuales una propuesta pasa por todas sus fases hasta ser ratificada o derogada.

En primer lugar, un cargo formula una propuesta en base a una necesidad, solución o mejora que considere oportuna para la comunidad. Esta formulación conlleva la redacción de un texto explicativo y detallado que contenga todo el desarrollo de la iniciativa. Además, deberá de incluir las opciones de voto que deberán ser dos o más. Las opciones de voto dependerán del tipo de propuesta, por ejemplo, para un cambio de

ascensor en una comunidad de vecinos, la propuesta deberá contener al menos dos opciones de voto con los presupuestos de las empresas de ascensores.

A continuación, la propuesta, en calidad de pre-propuesta, pasa a ser ratificada para votación por los otros cargos. Es decir, para que los miembros de una asamblea puedan votar sobre las opciones de voto, antes los demás cargos, deberán determinar si es una buena propuesta o no. Recordemos que los cargos se eligen de manera que sean representativos de la asamblea. Para superar la fase de pre-propuesta deberá ser ratificada por al menos la mitad de los cargos. Esta votación de ratificación o derogación se realizará mediante el uso del DNIe de manera que el proceso sea verificable.

Una vez la propuesta es aceptada para el voto comienza el período de quince días en el que todos los miembros (incluidos cargos) podrán votar usando el DNI electrónico las distintas opciones de voto de la propuesta. La fecha límite de votación de una propuesta (D_n) viene determinada por la fecha límite de la última propuesta (D_{n-1}):

- Si D_{n-1} más 3 días es menor a la fecha de aprobación más 15 días, entonces D_n será la fecha de aprobación más 15 días.
- En caso contrario la fecha límite responderá a la ecuación: $D_n = D_{n-1} + 3$.

Utilizando este control sobre las propuestas se evita que se acumulen más de 5 propuestas simultáneas en una asamblea, que podría provocar la saturación del usuario que tendría que separar su atención en demasiados asuntos.

Cuando se termina el periodo de votación de quince días se “cierran las urnas” y se procede al recuento. Al ser un sistema completamente informatizado los resultados se publican instantáneamente junto con estadísticas significativas de la votación.

3.2. Especificación de estándares y normas

Este apartado recoge los estándares, normas, leyes y recomendaciones que deben tenerse en cuenta para el desarrollo del proceso de análisis del sistema.

3.2.1. Restricciones Generales

Las restricciones generales son aquellas que afectan al sistema de información que queremos desarrollar.

- La interfaz de la aplicación deberá ser sencilla y amigable al usuario y sin ambigüedades.
- El sistema cliente se ejecutará exclusivamente sobre dispositivos móviles con sistema operativo Android desde la versión 4.0.3.
- El sistema servidor se ejecutará exclusivamente sobre el sistema operativo Ubuntu 16.04 LTS.

- El gestor de base de datos será MySQL versión 5.7.17.
- El entorno de desarrollo de software será Android Studio versión 2.2.3.
- Los lenguajes de programación para cliente serán Java y XML.
- El lenguaje de programación para servidor será Python 2.8 usando el framework Django.
- El idioma de interfaz será únicamente español (España).
- La aplicación deberá cumplir con el Acuerdo de Distribución para desarrolladores de Google Play.
- Requisitos mínimos del dispositivo móvil:
  - Sistema operativo Android 4.0.3.
  - Procesador 800 Mhz
  - WiFi 802.11 b/g/n
  - Interfaz Near Field Communication (NFC)
  - Memoria Interna de 2GB

3.2.2. Supuestos y dependencias

En este punto se describen los supuestos de los que parte el análisis del sistema y se especifican las dependencias con otros sistemas, procesos, legislaciones, etc.

El subsistema servidor debe ser agnóstico en cuanto a lenguajes de programación en su interfaz de comunicación. Además, deberá ser lo más modular y desacoplado posible. El acceso a todas las operaciones es concurrente, no se limita a un número de usuarios.

El subsistema cliente de Android deberá soportar operaciones con el DNIe y ser tolerante a fallos. Además, deberá tener una interfaz de usuario muy sencilla que prime usabilidad frente a diseño.

El proceso de registro, autenticación y voto deberá utilizar el DNIe y ser seguro, fiable y tolerante a fallos.

Los datos relativos a los usuarios recogidos en la base de datos deben cumplir con la Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal.

3.3. Establecimiento de Requisitos del Software

A lo largo de este apartado se plasmarán los requisitos de usuario para así determinar los requisitos de software del sistema. Para ello se seguirán las siguientes normas de estilo y redacción que posibiliten y faciliten la comprensión:

- **Sin ambigüedades**: Los requisitos deben tener un único sentido y significado.

- **Completitud**: Se recogerán todos los requisitos sin excepción.
- **Verificación**: Se podrá comprobar que se cumple con los requisitos.
- **Coherencia**: No debe existir ninguna contradicción entre los diferentes requisitos.
- **Modificable**: Los requisitos podrán estar sujetos a modificación.
- **Trazable**: Clara identificación del origen y consecuencias de los requisitos.
- **Utilizable**: Los requisitos serán fáciles de utilizar en el resto del documento.

Se comenzará con una especificación de los casos de uso y después se hará un análisis de los requisitos de software necesarios para que el sistema funcione correctamente y de acuerdo al objetivo final de este proyecto.

3.3.1. Especificación de Casos de Uso

Esta especificación engloba los diagramas y tablas que recogen los distintos casos de uso con el fin de facilitar la comprensión de las interacciones que el usuario puede realizar en el sistema.

Se comenzará por presentar un diagrama de casos de uso explicativo donde se relacionan los actores con las funcionalidades.

Un actor es un usuario externo que dota de funcionalidad al sistema y guarda relación con el mismo. Este sistema tiene tres tipos de actores:

- **Votante**: Es el actor base o padre del resto. Se relaciona con todos los casos de uso básicos del sistema.
- **Cargo**: Actor que hereda de Votante y se relaciona con los casos de uso especiales que dotan de funcionalidad al rol de Cargo en una Asamblea.
- **Administrador**: Actor que hereda de Votante y se relaciona con los casos de uso especiales que dotan de funcionalidad al rol de Administrador en una Asamblea.

3.3.1.1. Diagrama de casos de uso

El siguiente diagrama de casos de uso recoge todas las relaciones entre casos de uso y actores:

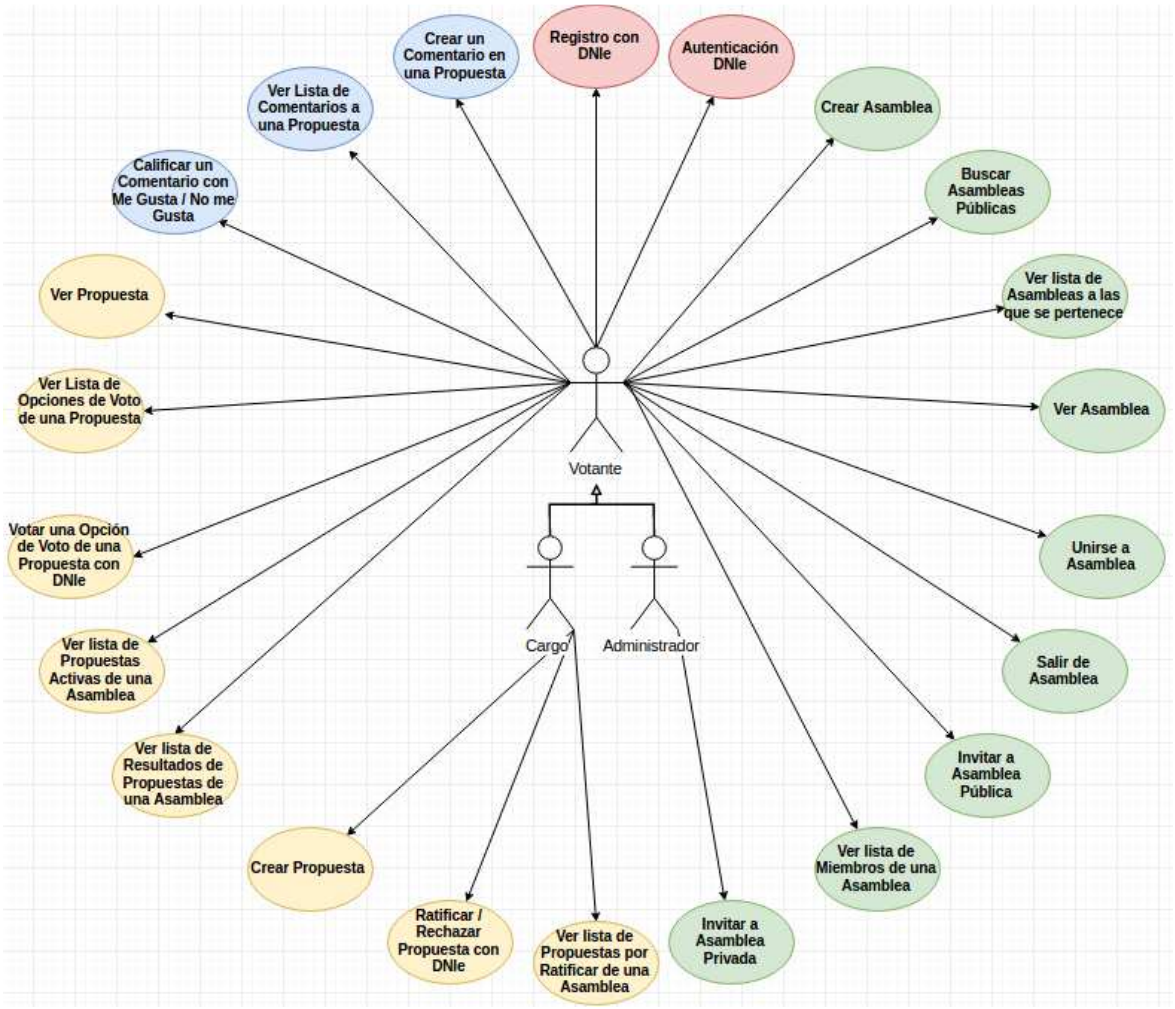


Ilustración 2: Casos de uso

3.3.1.2. Definición de casos de uso

A continuación, se definirán y detallarán formalmente los casos de uso del sistema. La siguiente tabla servirá para recoger ordenadamente los distintos casos de uso:

Identificador	
Nombre	
Actor	
Objetivo	
Condiciones Previas	
Condiciones Posteriores	
Escenario	

Tabla 1: Tabla base para casos de uso

Donde:

- **Identificador**: Código identificador unívoco de cada caso de uso. Será de la forma CU-XXX donde XXX serán valores enteros desde 000 hasta 999.
- **Nombre**: Texto descriptivo del caso de uso.
- **Actor**: Usuario que puede realizar el caso de uso.
- **Objetivo**: Finalidad del caso de uso.
- **Condiciones previas**: Condiciones que se deben cumplir para poder realizar el caso de uso.
- **Condiciones posteriores**: Estado del sistema después de realizar el caso de uso.
- **Escenario**: Descripción de la ejecución del caso de uso paso a paso.

Identificador	CU-001
Nombre	Registro con DNIe
Actor	Votante
Objetivo	Permitir la inserción de los datos del usuario en la aplicación para la creación de una cuenta mediante el uso del DNIe
Condiciones Previas	• Poseer DNIe v3.0 • Poseer Smartphone Android con NFC • Tener descargada e instalada la aplicación
Condiciones Posteriores	• Se introducirá al usuario en la base de datos con sus datos personales obtenidos a través del DNI y de un formulario complementario (email, aceptación de T&C).
Escenario	Se realizará una única vez cuando el usuario entre por primera vez en la aplicación.

Tabla 2: CU-001, Registro con DNIe

Identificador	CU-002
Nombre	Autenticación con DNIe
Actor	Votante
Objetivo	Validar la entrada de un usuario a la aplicación si aún no ha entrado.
Condiciones Previas	• Poseer DNIe v3.0 • Poseer Smartphone Android con NFC • Tener descargada e instalada la aplicación
Condiciones Posteriores	• Se dará acceso al usuario al resto de funcionalidad. • Se grabarán datos de acceso al servidor en el dispositivo cliente para una mejor experiencia de usuario
Escenario	Se realizará cada vez que no existan datos guardados en el dispositivo cliente relativos a la autenticación del usuario.

Tabla 3: CU-002, Autenticación con DNIe

Identificador	CU-003
Nombre	Crear Asamblea
Actor	Votante
Objetivo	Creación de un grupo de interés
Condiciones Previas	• Estar autenticado
Condiciones Posteriores	• Se creará una asamblea donde el usuario que la crea será administrador. • La asamblea tendrá unas elecciones a los 15 días, periodo en el que se podrán invitar o unirse otros usuarios (en caso de que la asamblea sea privada el administrador tendrá que invitar a los miembros)
Escenario	Se dará cada vez que un usuario desee crear una Asamblea

Tabla 4: CU-003, Crear asamblea

Identificador	CU-004
Nombre	Buscar Asamblea Pública
Actor	Votante
Objetivo	Localizar asambleas públicas de interés para el usuario
Condiciones Previas	• Estar autenticado
Condiciones Posteriores	• Se listarán los resultados de la búsqueda
Escenario	Cuando el usuario desee buscar una asamblea ya creada

Tabla 5: CU-004, Buscar asamblea pública

Identificador	CU-005
Nombre	Ver lista de Asambleas
Actor	Votante
Objetivo	Ver la lista de Asambleas de las que el usuario es miembro.
Condiciones Previas	• Estar autenticado
Condiciones Posteriores	• Se obtendrá la lista de Asambleas en las que el usuario es miembro.
Escenario	Cuando el usuario desee ver la lista de las Asambleas a las que pertenece.

Tabla 6: CU-005, Ver lista de asambleas

Identificador	CU-006
Nombre	Ver Asamblea
Actor	Votante
Objetivo	Ver Asamblea en la que el usuario es miembro
Condiciones Previas	• Estar autenticado • Ser miembro de la asamblea
Condiciones Posteriores	• Se obtendrá la Asamblea en la que el usuario es miembro
Escenario	Cuando el usuario desee ver una Asamblea en la que es miembro.

Tabla 7: CU-006, Ver asamblea

Identificador	CU-007
Nombre	Unirse a Asamblea
Actor	Votante
Objetivo	Permitir a un usuario ser miembro de una asamblea pública
Condiciones Previas	• Estar autenticado • Que la asamblea objetivo sea pública
Condiciones Posteriores	• El usuario se convierte en miembro de la asamblea
Escenario	Se dará cada vez que el usuario quiera pertenecer a una asamblea pública.

Tabla 8: CU-007, Unirse a asamblea

Identificador	CU-008
Nombre	Salir de Asamblea
Actor	Votante
Objetivo	Permitir a un usuario dejar de ser miembro de una asamblea
Condiciones Previas	• Estar autenticado • Ser miembro de la asamblea
Condiciones Posteriores	• El usuario deja de ser miembro de la asamblea
Escenario	Cuando un usuario quiera dejar de pertenecer a una asamblea

Tabla 9: CU-008, Salir de asamblea

Identificador	CU-009
Nombre	Invitar a Asamblea Pública
Actor	Votante
Objetivo	Permite a un usuario hacer miembro a otro usuario de una asamblea en la que es miembro
Condiciones Previas	• Estar autenticado • Ser miembro de la asamblea
Condiciones Posteriores	• El usuario objetivo se convierte en miembro de la asamblea
Escenario	Cuando un usuario quiera hacer partícipe a otro en una asamblea

Tabla 10: CU-009, Invitar a asamblea pública

Identificador	CU-010
Nombre	Ver lista de miembros de una Asamblea
Actor	Votante
Objetivo	Listar los miembros y su rol dentro de una asamblea
Condiciones Previas	• Estar autenticado • Ser miembro de la asamblea
Condiciones Posteriores	• Se muestra una lista de los miembros y el cargo que ostentan en la asamblea
Escenario	Cada vez que el usuario quiera ver quien se encuentra dentro de la asamblea

Tabla 11: CU-010, Ver lista de miembros de una asamblea

Identificador	CU-011
Nombre	Invitar a Asamblea privada
Actor	Administrador de Asamblea
Objetivo	Permitir al administrador de la asamblea privada que otro usuario pueda ser miembro de la misma.
Condiciones Previas	• Estar autenticado • Ser miembro de la asamblea • Ostentar rol de Administrador de Asamblea
Condiciones Posteriores	• El usuario objetivo pasa a ser miembro de la asamblea
Escenario	Cuando un Administrador desee añadir a un usuario a la asamblea privada.

Tabla 12: CU-011, Invitar a asamblea privada

Identificador	CU-012
Nombre	Ver lista de Propuestas por ratificar de una Asamblea
Actor	Cargo
Objetivo	Listar las Propuestas que aún no han sido aceptadas por los cargos para ser votadas.
Condiciones Previas	• Estar autenticado • Ser miembro de la asamblea • Ostentar rol de Cargo
Condiciones Posteriores	• Se listarán las Propuestas que aún no han sido aceptadas por los cargos para ser votadas
Escenario	Cuando un Cargo desee ver las propuestas aún no ratificadas

Tabla 13: CU-012, Ver lista de propuestas por ratificar de una asamblea

Identificador	CU-013
Nombre	Ratificar / Rechazar Propuesta con DNIe
Actor	Cargo
Objetivo	Permite a un cargo votar con el DNIe si aceptar o rechazar la votación de una propuesta.
Condiciones Previas	• Estar autenticado • Ser miembro de la asamblea • Ostentar rol de Cargo • Tener el DNIe v3.0
Condiciones Posteriores	• El usuario emite su voto de ratificación o rechazo
Escenario	Se producirá siempre que un Cargo quiera votar la ratificación o rechazo de una propuesta.

Tabla 14: CU-013, Ratificar/Rechazar propuesta con DNIe

Identificador	CU-014
Nombre	Crear Propuesta
Actor	Cargo
Objetivo	Crear una nueva propuesta en la asamblea
Condiciones Previas	• Estar autenticado • Ser miembro de la asamblea • Ostentar rol de Cargo
Condiciones Posteriores	• Se creará una nueva propuesta en la asamblea que quedará pendiente de ratificación o rechazo
Escenario	Cuando un Cargo decida crear una propuesta para ser votada por la comunidad.

Tabla 15: CU-014, Crear propuesta

Identificador	CU-015
Nombre	Ver lista de Resultados de Propuestas de una Asamblea
Actor	Votante
Objetivo	Consultar los resultados de las votaciones a las Propuestas en una Asamblea
Condiciones Previas	• Estar autenticado • Ser miembro de la asamblea
Condiciones Posteriores	• Se muestra una lista de las propuestas terminadas con los resultados de todas las opciones
Escenario	Se produce cuando un usuario quiere ver las propuestas terminadas

Tabla 16: CU-015, Ver lista de resultados de propuestas de una asamblea

Identificador	CU-016
Nombre	Ver lista de Propuestas Activa de una Asamblea
Actor	Votante
Objetivo	Consultar las lista de las Propuestas que están en periodo de votación para los Votantes
Condiciones Previas	• Estar autenticado • Ser miembro de la asamblea
Condiciones Posteriores	• Se muestra una lista de las propuestas activas con el porcentaje de participación y la fecha en la que se termina la votación.
Escenario	Se produce cuando un usuario quiere ver las propuestas activas

Tabla 17: CU-016, Ver lista de propuestas activas de una asamblea

Identificador	CU-017
Nombre	Votar una opción de una Propuesta con DNIe
Actor	Votante
Objetivo	Permite a un votante votar con el DNIe una de las opciones de la Propuesta
Condiciones Previas	• Estar autenticado • Ser miembro de la asamblea • Tener el DNIe v3.0
Condiciones Posteriores	• El votante emite un voto con la opción de su elección
Escenario	Se produce cuando un usuario quiere votar una de las opciones de una propuesta

Tabla 18: CU-017, Votar una opción de una propuesta con DNIe

Identificador	CU-018
Nombre	Ver lista de Opciones de voto de una Propuesta
Actor	Votante
Objetivo	Permite al usuario ver la lista de opciones que ofrece una Propuesta.
Condiciones Previas	• Estar autenticado • Ser miembro de la asamblea
Condiciones Posteriores	• Se muestra la lista de opciones que ofrece una Propuesta
Escenario	Se da cuando un usuario quiere ver las opciones que ofrece una propuesta

Tabla 19: CU-018, Ver lista de opciones de voto de una propuesta

Identificador	CU-019
Nombre	Ver Propuesta
Actor	Votante
Objetivo	Permite ver una propuesta
Condiciones Previas	• Estar autenticado • Ser miembro de la asamblea
Condiciones Posteriores	• Se muestra la propuesta
Escenario	Se produce cuando quiere ver una única propuesta

Tabla 20: CU-019, Ver Propuesta

Identificador	CU-020
Nombre	Calificar un Comentario
Actor	Votante
Objetivo	Permite a un usuario votar si un comentario le gusta o no
Condiciones Previas	• Estar autenticado • Ser miembro de la asamblea • Que exista el comentario
Condiciones Posteriores	• Se emite la votación del comentario y se actualiza la reputación del usuario propietario del comentario.
Escenario	Se produce cuando a un usuario que le gusta o no un comentario decide votar dicho comentario.

Tabla 21: CU-020, Calificar un comentario

Identificador	CU-021
Nombre	Ver lista de Comentarios a una Propuesta
Actor	Votante
Objetivo	Permite a un usuario ver una lista de los comentarios que se han hecho a una Propuesta
Condiciones Previas	• Estar autenticado • Ser miembro de la asamblea
Condiciones Posteriores	• Se muestra la lista de los comentarios que se han hecho en la Propuesta
Escenario	Cuando un usuario quiera ver los comentarios que se han hecho en una Propuesta

Tabla 22: CU-021, Ver lista de comentarios a una propuesta

Identificador	CU-022
Nombre	Crear un Comentario en una Propuesta
Actor	Votante
Objetivo	Permitir al usuario opinar o comentar sobre una Propuesta o responder a un comentario de otro usuario.
Condiciones Previas	• Estar autenticado • Ser miembro de la asamblea
Condiciones Posteriores	• Se crea un comentario a la Propuesta
Escenario	Cuando un usuario quiera opinar, comentar o responder a otro comentario en una Propuesta.

Tabla 23: CU-022, Crear un comentario en una propuesta

3.3.2. Obtención de requisitos

En este apartado se detallarán los requisitos de software para el sistema derivados de los requisitos y los casos de uso.

Los requisitos se dividirán en dos grandes grupos:

- Funcionales: o requisitos de capacidad. Indican las funcionalidades y necesidades del sistema.
- No funcionales: o de restricción. Limitan capacidades en el desarrollo del proyecto.

Los requisitos de software seguirán el siguiente formato:

Identificador	RS[Y]-[XXX]
Título	
Descripción	
Prioridad	[Alta, Media, Baja]
Estabilidad	[Sí, No]
Necesidad	[Esencial, Deseable, Opcional]

Tabla 24: Tabla base para requisitos

Dónde:

- **Identificador**: Código único identificador del requisito. RS hace referencia a Requisito de Software, [Y] tendrá valor F para requisitos funcionales y NF para requisitos no funcionales. El orden no es representativo de la importancia del requisito.
- **Título**: Texto que indica el nombre del requisito.
- **Descripción**: Explicación breve de la funcionalidad o restricción que representa el requisito.
- **Prioridad**: Una prioridad más alta indica que ese requisito debe ser tratado con mayor urgencia.
- **Estabilidad**: Indica si el requisito es más fácilmente susceptible a cambios o no.
- **Necesidad**: Representa la importancia del requisito para el proyecto. Los “esenciales” han de cumplirse siempre, los “deseables” es óptimo que se cumplan, los “opcionales” no alteran el resultado final del proyecto

3.3.2.1. Requisitos funcionales

Identificador	RSF-001
Título	Autenticación
Descripción	Los usuarios podrán autenticarse con el DNIe.
Prioridad	Alta
Estabilidad	Sí
Necesidad	Esencial

Tabla 25: RSF-001, Autenticación

Identificador	RSF-002
Título	Registro
Descripción	Los usuarios podrán registrarse con el DNIe.
Prioridad	Alta
Estabilidad	Sí
Necesidad	Esencial

Tabla 26: RSF-002, Registro

Identificador	RSF-003
Título	Formulario de registro
Descripción	El usuario deberá introducir su Email para crearse una cuenta.
Prioridad	Media
Estabilidad	Sí
Necesidad	Esencial

Tabla 27: RSF-003, Formulario de registro

Identificador	RSF-004
Título	Condiciones de uso
Descripción	El usuario deberá aceptar las condiciones de uso para crearse una cuenta.
Prioridad	Alta
Estabilidad	Sí
Necesidad	Esencial

Tabla 28: RSF-004, Condiciones de uso

Identificador	RSF-005
Título	Inicio de sesión
Descripción	La aplicación permitirá iniciar sesión a los usuarios con el DNIe.
Prioridad	Alta
Estabilidad	Sí
Necesidad	Esencial

Tabla 29: RSF-005, Inicio de sesión

Identificador	RSF-006
Título	Cierre de sesión
Descripción	La aplicación permitirá cerrar sesión a los usuarios.
Prioridad	Alta
Estabilidad	Sí
Necesidad	Esencial

Tabla 30: RSF-006, Cierre de sesión

Identificador	RSF-007
Título	Crear asamblea
Descripción	Los usuarios podrán crear una asamblea mediante un formulario con los siguientes campos: • Nombre • Zona de actuación • Visibilidad (pública o privada) • Descripción
Prioridad	Alta
Estabilidad	Sí
Necesidad	Esencial

Tabla 31: RSF-007, Crear asamblea

Identificador	RSF-008
Título	Buscar asamblea
Descripción	Los usuarios podrán buscar asambleas por el nombre
Prioridad	Media
Estabilidad	Sí
Necesidad	Esencial

Tabla 32: RSF-008, Buscar asamblea

Identificador	RSF-009
Título	Unirse a asamblea
Descripción	Los usuarios podrán unirse a asambleas previamente creadas
Prioridad	Alta
Estabilidad	Sí
Necesidad	Esencial

Tabla 33: RSF-009, Unirse a asamblea

Identificador	RSF-010
Título	Invitar a asamblea pública
Descripción	Los usuarios miembros de una asamblea pública podrán invitar a otros usuarios a la asamblea.
Prioridad	Baja
Estabilidad	Sí
Necesidad	Esencial

Tabla 34: RSF-010, Invitar a asamblea pública

Identificador	RSF-011
Título	Invitar a asamblea privada
Descripción	El usuario con el rol “Administrador” de una asamblea podrá invitar a otros usuarios a la asamblea.
Prioridad	Baja
Estabilidad	Sí
Necesidad	Esencial

Tabla 35: RSF-011, Invitar a asamblea privada

Identificador	RSF-012
Título	Salir de asamblea
Descripción	Los usuarios podrán salir de las asambleas de las que son miembros
Prioridad	Alta
Estabilidad	Sí
Necesidad	Esencial

Tabla 36: RSF-012, Salir de asamblea

Identificador	RSF-013
Título	Lista de asambleas
Descripción	Los usuarios podrán ver un listado de las asambleas de las que son miembros.
Prioridad	Alta
Estabilidad	Sí
Necesidad	Esencial

Tabla 37: RSF-013, Lista de asambleas

Identificador	RSF-014
Título	Ver asamblea
Descripción	Los usuarios podrán ver los detalles de una asamblea.
Prioridad	Alta
Estabilidad	Sí
Necesidad	Esencial

Tabla 38: RSF-014, Ver asamblea

Identificador	RSF-015
Título	Ver lista de pre-propuestas
Descripción	Los usuarios que tengan el rol de “Cargo” podrán ver la lista de pre-propuestas de una asamblea
Prioridad	Alta
Estabilidad	Sí
Necesidad	Deseable

Tabla 39: RSF-015, Ver lista de pre-propuestas

Identificador	RSF-016
Título	Ver lista de propuestas activas
Descripción	Los usuarios podrán ver la lista de propuestas activas de una asamblea
Prioridad	Alta
Estabilidad	Sí
Necesidad	Deseable

Tabla 40: RSF-016, Ver lista de propuestas activas

Identificador	RSF-017
Título	Ver lista de resultados de propuestas terminadas
Descripción	Los usuarios podrán ver la lista de propuestas activas de una asamblea y sus resultados
Prioridad	Alta
Estabilidad	Sí
Necesidad	Deseable

Tabla 41: RSF-017, Ver lista de resultados de propuestas terminadas

Identificador	RSF-018
Título	Ver lista de miembros
Descripción	Los usuarios podrán ver la lista de miembros de la asamblea, sus respectivos roles y reputación
Prioridad	Alta
Estabilidad	Sí
Necesidad	Deseable

Tabla 42: RSF-018, Ver lista de miembros

Identificador	RSF-019
Título	Crear propuesta
Descripción	Los usuarios con el rol de “Cargo” podrán crear una propuesta a través de un formulario con los siguientes campos: • Título de la propuesta • Texto de la propuesta • Lista de opciones de voto
Prioridad	Alta
Estabilidad	Sí
Necesidad	Esencial

Tabla 43: RSF-019, Crear propuesta

Identificador	RSF-020
Título	Ver propuesta
Descripción	Los usuarios podrán ver los detalles de una propuesta
Prioridad	Alta
Estabilidad	Sí
Necesidad	Esencial

Tabla 44: RSF-020, Ver propuesta

Identificador	RSF-021
Título	Ver lista de comentarios
Descripción	Los usuarios podrán ver la lista de comentarios de una propuesta
Prioridad	Media
Estabilidad	Sí
Necesidad	Deseable

Tabla 45: RSF-021, Ver lista de comentarios

Identificador	RSF-022
Título	Crear comentario
Descripción	Los usuarios podrán crear comentarios en las propuestas
Prioridad	Media
Estabilidad	Sí
Necesidad	Deseable

Tabla 46: RSF-022, Crear comentario

Identificador	RSF-023
Título	Votar comentario
Descripción	Los usuarios podrán votar con “Me gusta”/”No me gusta” a los comentarios
Prioridad	Media
Estabilidad	Sí
Necesidad	Deseable

Tabla 47: RSF-023, Votar comentario

Identificador	RSF-024
Título	Sistema de reputación
Descripción	Mediante los votos a comentarios los usuarios obtendrán o perderán reputación en la asamblea.
Prioridad	Media
Estabilidad	Sí
Necesidad	Deseable

Tabla 48: RSF-024, Sistema de reputación

Identificador	RSF-025
Título	Lista de opciones de voto de pre-propuesta
Descripción	Los usuarios con el rol de “Cargo” podrán ver la lista con las opciones “Aceptar propuesta” y “Rechazar propuesta”.
Prioridad	Alta
Estabilidad	Sí
Necesidad	Esencial

Tabla 49: RSF-025, Lista de opciones de voto de pre-propuesta

Identificador	RSF-026
Título	Lista de opciones de voto de propuesta activa
Descripción	Los usuarios podrán ver la lista de opciones de voto de una propuesta activa.
Prioridad	Alta
Estabilidad	Sí
Necesidad	Esencial

Tabla 50: RSF-026, Lista de opciones de voto de propuesta activa

Identificador	RSF-027
Título	Lista de opciones de voto de propuesta terminada
Descripción	Los usuarios podrán ver la lista de opciones de voto de una propuesta terminada con sus resultados.
Prioridad	Alta
Estabilidad	Sí
Necesidad	Esencial

Tabla 51: RSF-027, Lista de opciones de voto de propuesta terminada

Identificador	RSF-028
Título	Ratificar / Rechazar Propuesta con DNIe
Descripción	Los usuarios con el rol “Cargo” podrán votar con el DNIe una de las opciones: “Aceptar propuesta” o “Rechazar propuesta”
Prioridad	Alta
Estabilidad	Sí
Necesidad	Esencial

Tabla 52: RSF-028, Ratificar / Rechazar propuesta con DNIe

Identificador	RSF-029
Título	Voto con DNIe
Descripción	Los usuarios podrán votar las opciones de una propuesta
Prioridad	Alta
Estabilidad	Sí
Necesidad	Esencial

Tabla 53: RSF-029, Voto con DNIe

Identificador	RSF-030
Título	Sistema de elección de cargos
Descripción	Los miembros de una asamblea serán elegidos aleatoriamente para ocupar el rol de “Cargo” si su reputación es mayor o igual a la media de todos los miembros de la asamblea.
Prioridad	Alta
Estabilidad	Sí
Necesidad	Esencial

Tabla 54: RSF-030, Sistema de elección de cargos

3.3.2.2. Requisitos no funcionales

Identificador	RSNF-001
Título	Rendimiento
Descripción	El sistema soportará concurrentemente hasta 1.000.000 usuarios
Prioridad	Media
Estabilidad	Sí
Necesidad	Deseable

Tabla 55: RSNF-001, Rendimiento

Identificador	RSNF-002
Título	Disponibilidad
Descripción	El sistema estará operativo 365 días al año.
Prioridad	Alta
Estabilidad	Sí
Necesidad	Deseable

Tabla 56: RSNF-002, Disponibilidad

Identificador	RSNF-003
Título	Accesibilidad
Descripción	El sistema será accesible a personas con discapacidad visual.
Prioridad	[Alta, Media, Baja]
Estabilidad	Sí
Necesidad	Deseable

Tabla 57: RSNF-003, Accesibilidad

Identificador	RSNF-004
Título	Documento Nacional de Identidad
Descripción	El usuario deberá poseer el Documento Nacional de Identidad versión 3.0
Prioridad	Alta
Estabilidad	Sí
Necesidad	Esencial

Tabla 58: RSNF-004, Documento Nacional de Identidad

Identificador	RSNF-005
Título	Contraseña Documento Nacional de Identidad
Descripción	El usuario deberá conocer la contraseña del Documento Nacional de Identidad versión 3.0
Prioridad	Alta
Estabilidad	Sí
Necesidad	Esencial

Tabla 59: RSNF-005, Contraseña Documento Nacional de Identidad

Identificador	RSNF-006
Título	Dispositivos soportados
Descripción	La aplicación será soportada por cualquier dispositivo con un sistema operativo Android con una versión superior a 4.0.3
Prioridad	Alta
Estabilidad	Sí
Necesidad	Esencial

Tabla 60: RSNF-006, Dispositivos soportados

Identificador	RSNF-007
Título	NFC
Descripción	El dispositivo Android deberá tener interfaz NFC para la comunicación con el DNIe
Prioridad	Alta
Estabilidad	Sí
Necesidad	Esencial

Tabla 61: RSNF-007, NFC

Identificador	RSNF-008
Título	Idioma
Descripción	La aplicación solo será desarrollada en lenguaje Español de España
Prioridad	Media
Estabilidad	Sí
Necesidad	Deseable

Tabla 62: RSNF-008, Idioma

Identificador	RSNF-09
Título	Escalabilidad
Descripción	El sistema se desarrollará de manera que se pueda escalar con facilidad.
Prioridad	Media
Estabilidad	Sí
Necesidad	Deseable

Tabla 63: RSNF-009, Escalabilidad

Identificador	RSNF-010
Título	Interfaz
Descripción	La interfaz de usuario será sencilla y en conformidad con la guía de estilo de Android
Prioridad	Media
Estabilidad	Sí
Necesidad	Deseable

Tabla 64: RSNF-010, Interfaz

Identificador	RSNF-011
Título	SSL
Descripción	Las conexiones entre la aplicación y el servidor estarán cifradas mediante SSL/TLS
Prioridad	Alta
Estabilidad	Sí
Necesidad	Deseable

Tabla 65: RSNF-011, SSL

Identificador	RSNF-012
Título	SSL
Descripción	Las conexiones entre la aplicación y el servidor estarán cifradas mediante SSL/TLS
Prioridad	Alta
Estabilidad	Sí
Necesidad	Deseable

Tabla 66: RSNF-012

3.3.3. Matriz de trazabilidad

La matriz de trazabilidad es una herramienta usada para asegurar que los requisitos aportan valor. Esta herramienta se formula en forma de tabla en la que se muestran las relaciones de los requisitos con su origen. De esta manera se puede hacer un seguimiento durante el proyecto.

La siguiente tabla muestra la matriz de trazabilidad de este proyecto:

	CU-001	CU-002	CU-003	CU-004	CU-005	CU-006	CU-007	CU-008	CU-009	CU-010	CU-011	CU-012	CU-013	CU-014	CU-015	CU-016	CU-017	CU-018	CU-019	CU-020	CU-021	CU-022
RSF-001		X																				
RSF-002	X																					
RSF-003	X																					
RSF-004	X																					
RSF-005		X																				
RSF-006																						
RSF-007			X																			
RSF-008				X																		
RSF-009							X															
RSF-010									X													
RSF-011											X											
RSF-012								X														
RSF-013					X																	
RSF-014						X																
RSF-015												X										
RSF-016																X						
RSF-017															X							
RSF-018										X												
RSF-019														X								
RSF-020																			X			
RSF-021																					X	
RSF-022																						X
RSF-023																				X		
RSF-024																						
RSF-025																		X				
RSF-026																		X				
RSF-027																		X				
RSF-028													X									
RSF-029																	X					
RSF-030																						

Tabla 67: Matriz de trazabilidad

4. Diseño

En este apartado se definirá el funcionamiento global del sistema mediante el uso de diagramas UML y la descripción del entorno operacional. Se utilizará la estructura del modelo de vistas de arquitectura 4+1 para describir los sistemas de software.

4.1. Vista lógica

La vista lógica describe la estructura y funcionalidad del sistema. Para ello, se utilizan diagramas UML como los diagramas de clases.

El diagrama de clases tiene como objetivo definir las distintas entidades con relevancia en el sistema, así como las relaciones entre ellas. De esta manera, más tarde se podrán definir adecuadamente estas relaciones que hacen posible una buena adecuación a la funcionalidad a implementar.

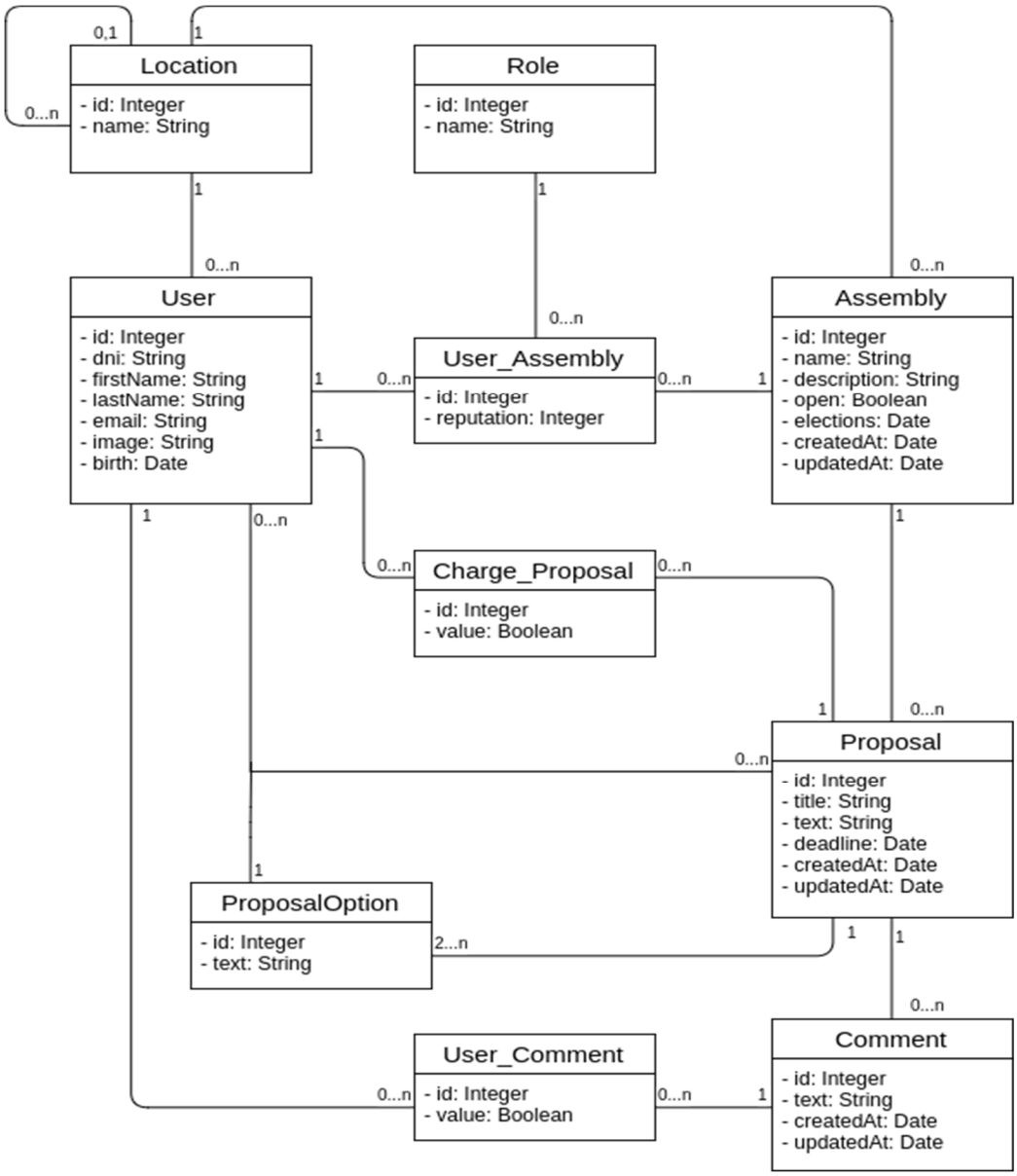


Ilustración 3: Diagrama de clases

Como podemos ver en la figura, hay siete entidades importantes:

- **User**: El usuario representa la identidad digital que tendrá una persona dentro de la aplicación. Contiene todos los atributos necesarios para el correcto funcionamiento de la aplicación. Estos datos se deberían extraer del DNI (excepto el email).
- **Assembly**: Esta entidad representa el concepto de espacio virtual donde se reúnen los usuarios con unos mismos intereses para debatir, votar y aprobar propuestas.
- **Role:** El rol representa el tipo de actor que es un usuario dentro de una asamblea. Es por ello que entre *User*, *Assembly* y Role exista una relación ternaria que define qué rol tiene un usuario en una asamblea. Los roles son cerrados, pero se ha decidido crear una entidad para ello por si posteriormente se desean añadir más.
- **Location**: Esta entidad representa las zonas de influencia de los usuarios y las asambleas. Esta entidad se autoreferencia porque puede pertenecer a otra zona de influencia. Es decir, si un usuario es de la ciudad de Toledo, sus zonas de influencia serán Toledo, la provincia de Toledo, Castilla La-Mancha y España. Estas zonas de influencia serán claves a la hora de que un usuario pueda o no ser miembro de una asamblea.
- **Proposal**: Una propuesta representa el documento con el cual un cargo puede comenzar una votación. La propuesta consta de un título, un texto descriptivo y una fecha de finalización de votación.
- **ProposalOption**: Esta entidad hace alusión a las diferentes opciones de votación dentro de una propuesta. Como mínimo tiene que haber dos opciones de voto.
- **Comment**: Los comentarios son textos que los usuarios pueden escribir para dar su opinión o debatir con otros usuarios dentro de una propuesta. Es decir, funciona a modo de foro ciudadano en el que los miembros de la asamblea pueden opinar.

El resto de entidades son secundarias y son fruto de las relaciones entre las entidades principales ya que estas relaciones tienen atributos que las definen.

4.2. Vista de desarrollo

La vista de desarrollo se enfoca en la administración de los artefactos de software desde la vista del programador. Para realizar esta tarea se utilizan diagramas de componentes.

El diagrama de componentes tiene como objetivo el modelado y relación de sistemas y subsistemas que más tarde serán implementadas. Este tipo de diagrama entra dentro de los diagramas UML y permite mostrar la arquitectura del sistema a alto nivel.

El siguiente diagrama expone los dos grandes componentes del sistema, así como sus subcomponentes. También se pueden ver las dependencias entre ellos.

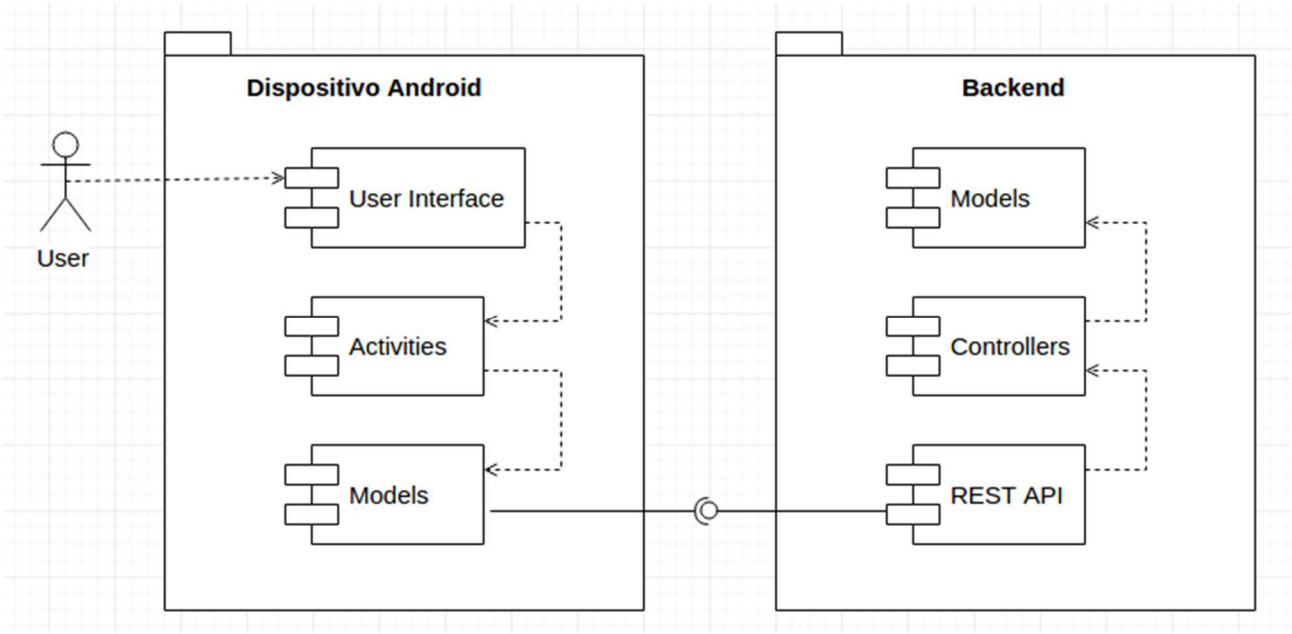


Ilustración 4: Diagrama de componentes

La arquitectura de la figura muestra claramente el modelo Cliente-Servidor en el cual el cliente sería el dispositivo Android y el servidor el componente anotado como “Backend”.

Los componentes y subcomponentes del sistema son:

- **Dispositivo Android**: Es el lado cliente del sistema, se encarga de hacer de intermediario entre la lógica de servidor y el usuario. Recoge las operaciones del usuario, las transforma en peticiones al servidor, recoge el resultado de la transacción y reacciona a este resultado.
    - *User Interface*: Es la parte con la que el usuario puede interactuar, tanto visualmente como pulsando sobre la pantalla.
    - *Activities*: Son los presentadores de información en la interfaz de usuario. Hacen de intermediarios entre el modelo y la interfaz. Se encargan de manejar la interacción del usuario y ofrecer una respuesta a la misma.
    - *Models*: Encapsulan la lógica del cliente. Se encarga principalmente de agrupar las distintas operaciones necesarias para la comunicación con el sistema Backend.
- **Backend**: Es el lado servidor del sistema, se encarga de toda la lógica de negocio necesaria para el correcto funcionamiento del sistema. Es agnóstica en cuanto al lenguaje del cliente y ofrece interfaces REST para la comunicación mediante JSON o XML.

- *REST API*: Es el subsistema calificado como Vista. Encargado de la comunicación con el exterior. Se encarga de recibir las operaciones con los datos pertinentes desde cualquier cliente con credenciales.
- *Controllers*: Son los controladores que se encargan de hacer de intermediarios entre la capa de Vista y la de Modelo.
- *Models*: Encapsulan toda la lógica de negocio del sistema. Son los encargados de realizar las operaciones necesarias de una manera segura y eficiente.

4.3. Vista de proceso

La vista de proceso engloba los aspectos necesarios para comprender como funcionan ciertos aspectos del sistema a lo largo de su ejecución. Para ello se utilizan los diagramas de secuencia.

Los diagramas de secuencia tienen como objetivo mostrar las interacciones entre los objetos de una aplicación a lo largo del tiempo. La figura de abajo muestra el diagrama de secuencia del proceso de voto mediante DNIe:

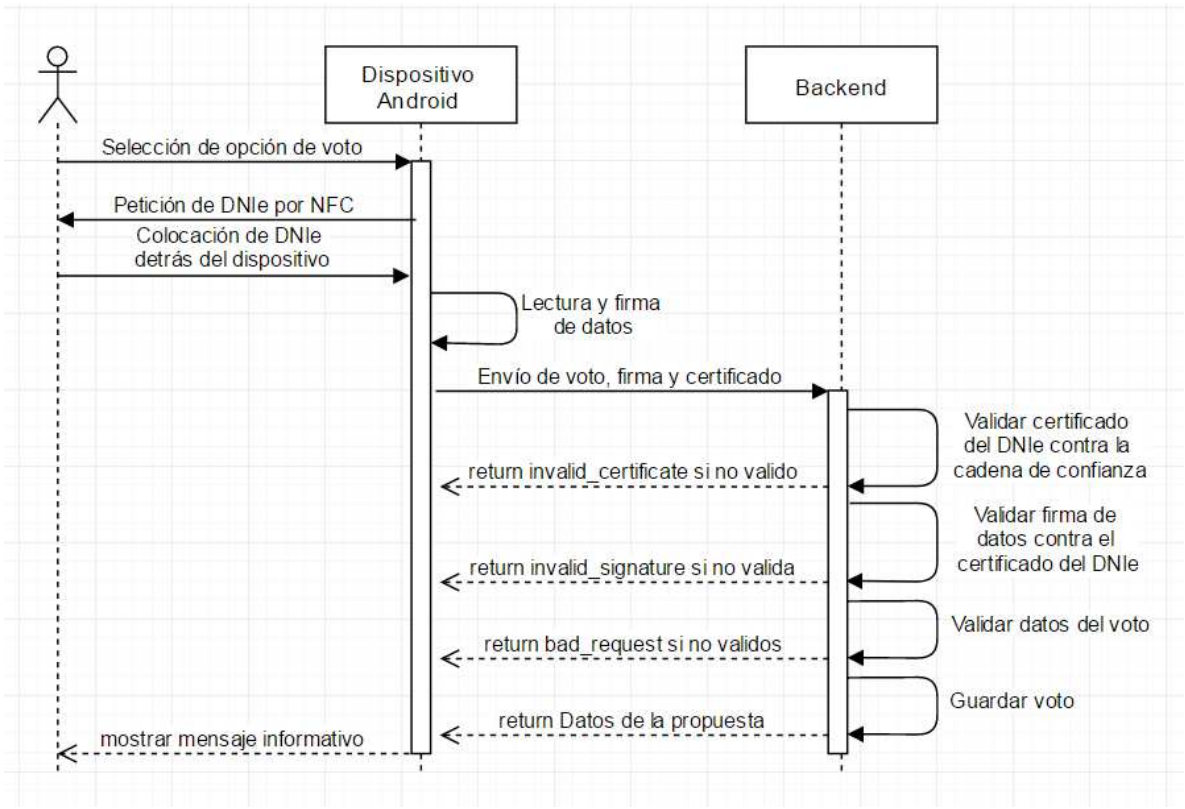


Ilustración 5: Diagrama de secuencia de voto

Este diagrama muestra como el usuario, al elegir una opción de voto, le es requerido que utilice el DNIe con el cual el dispositivo Android realizará la lectura y la firma. Después el dispositivo enviará los datos del voto, la firma del voto y el certificado público de firma al servidor para realizar la operación de voto. En el servidor se procederá a validar el certificado contra la cadena de confianza, después se validará la firma del

voto y a continuación se validarán los datos del voto, finalmente se guardará el voto. El servidor responderá con los datos de la propuesta votada y el usuario verá un mansaje informativo.

4.4. Vista física

La vista física intenta describir la topología del sistema tal y como la entendería un ingeniero de sistemas. También se conoce como vista de despliegue y se utilizan diagramas de despliegue para plasmarla gráficamente.

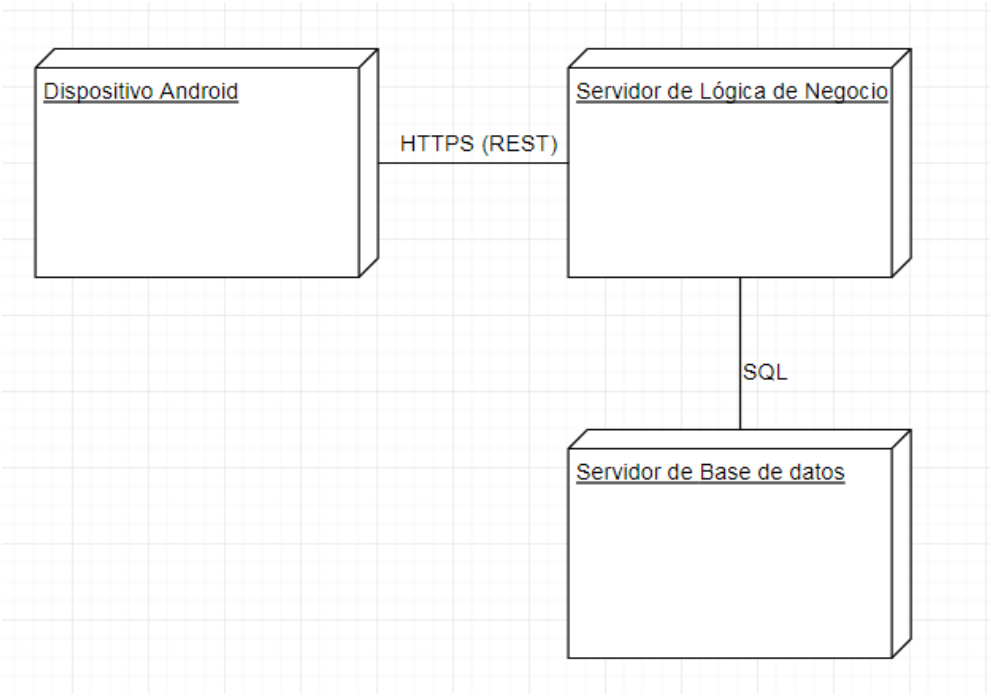


Ilustración 6: Diagrama de despliegue

El anterior diagrama de despliegue muestra como se ha diseñado el entorno de los componentes software desde la vista de un ingeniero de sistemas. Como se puede ver hay tres componentes:

- **Dispositivo Android**: Es el terminal de los usuarios que utilizan la aplicación. Contendrá la lógica de cliente y la interfaz de usuario. Este componente se comunicará con el Servidor de Lógica de negocio usando REST mediante HTTPS.
- **Servidor de Lógica de Negocio**: Este servidor se encarga de ejecutar la lógica de negocio del sistema usando el framework Django. Se comunica con el Servidor de Base de Datos mediante SQL.
- **Servidor de Base de Datos**: Este servidor contiene la parte del modelo que se encarga del almacenamiento de los datos. En este proyecto será el gestor de bases de datos MySQL.

4.5. Entorno operacional

En este proyecto se utiliza una arquitectura Cliente-Servidor que, por su modularidad, se puede establecer dentro de una arquitectura bastante desacoplada.

La arquitectura Cliente-Servidor permite una separación de lógicas que hacen posible un desarrollo paralelo e independiente. Además, si se intenta realizar una modularización del sistema, puede revertir en un desacoplamiento que permitirá la utilización de microservicios.

En este proyecto, a pesar de tener una alta modularidad, no se ha podido utilizar la arquitectura de microservicios ya que el coste en tiempo era bastante alto debido a la complejidad del sistema de votación en cuanto a seguridad y funcionalidad. Finalmente, se ha optado por un sistema monolítico modular.

De esta manera el entorno operacional sigue el siguiente diseño:

Ilustración 7: Entorno operacional

Las tecnologías utilizadas en estos componentes han sido:

- **Android**: La aplicación cliente se ha desarrollado para el sistema operativo Android. Este sistema operativo está basado en el núcleo de Linux y fue diseñado específicamente para dispositivos móviles con pantalla táctil. Se ha elegido esta tecnología ya que para hacer un cliente móvil es la que permite una comunicación con el chip NFC sin restricciones. Por ejemplo, el sistema operativo iOS no permite el acceso a este chip y por lo tanto no es un candidato viable.

- **Django**: La aplicación servidor se ha desarrollado usando este framework que provee de herramientas y métodos de trabajo que afianzan la seguridad del sistema. Se hablará más adelante sobre las características de este framework que lo hace ideal para la implementación rápida de servicios web.

- **MySQL**: Es la base de datos relacional más popular del mundo ya que ofrece una gestión y servicio de datos rápida a la vez que es gratuita y sencilla de manejar. Esta base de datos se usa en multitud de sitios web grandes como Wikipedia, Google, Facebook, etc.

A continuación, se describirá el entorno tecnológico que cubre las necesidades del sistema.

- Servidor AWS t2.micro:
  - CPU: 1 Intel Xeon alta frecuencia.
  - Memoria: 1 GB
  - Almacenamiento: EBS
- Aplicación Android desarrollada con Android Studio
  - Versión: 2.2.3
  - Lenguaje: Java, XML
  - Género: IDE
  - Sistema Operativo: Linux
- Documentación:
  - OpenOffice 5.1.6.2
  - Microsoft Office 2010
  - Inkscape
  - Gimp
  - Recursos multimedia online

5. Implementación y pruebas

Este apartado recogerá todo lo referente a cómo se ha llevado a cabo la implementación del sistema. Se comenzará por una explicación detallada de la implementación y las pruebas del lado servidor, después se pasará al lado cliente y finalmente a la integración de ambos.

5.1. Servidor

El subsistema servidor se encarga del almacenamiento de los datos, la lógica de negocio del sistema y de proveer servicios al exterior.

5.1.1. Implementación

Para realizar la implementación del servidor se ha utilizado el framework Django, puesto que provee de herramientas útiles para el desarrollo rápido, eficiente y seguro de aplicaciones web. Este framework sigue el patrón de diseño Modelo Vista Controlador y tiene cualidades muy interesantes como el mapeo de objetos relacionales, una API de base de datos robusta y una alta modularidad.

La base de datos utilizada es MySQL ya que ofrece las características necesarias de rendimiento, integración con Django y ofrece una gran robustez relacional.

Antes de comenzar con el desarrollo, hubo que crear un entorno virtual de Python con las librerías necesarias para Django, librerías necesarias para el manejo de certificados y librerías para crear servicios REST.

Los servicios implementados responden a una arquitectura Representational State Transfer (REST) los cuales servirán para la comunicación entre clientes y servidor.

La aplicación web se ha dividido en tres módulos funcionales: usuarios, asambleas y comentarios. Esta división responde al principio de modularidad y escalabilidad que en un futuro permitirá la inclusión de nueva funcionalidad de una manera más eficiente y con menos conflictos.

5.1.1.1. Módulo Usuarios

El módulo de usuarios contiene la lógica de negocio relativa a la autenticación que atañe a este TFG. Como la pretensión es integrar una autenticación mediante certificados con la librería de OAuth, se tuvo que adecuar el modelo de usuario que provee Django a uno con atributos propios para poder unificar la funcionalidad de la librería con el almacenamiento de datos relativos al DNI del usuario.

La siguiente figura muestra el código de la clase que representa la entidad de usuario. Esta clase hereda de AbstractUser que es la clase que utilizan los métodos de autenticación del framework. Más tarde se añadiría esta clase a la configuración del sistema para que el framework pudiera utilizarla en vez de la que usa por defecto.

```python
class User(AbstractUser):
    image = models.ImageField(upload_to='images')
    birth = models.DateField()
    city = models.ForeignKey(Location)
    assemblies = models.ManyToManyField(Assembly, through='UserAssembly', blank=True, default=[])
    votes = models.ManyToManyField(Proposal, through='UserProposal', blank=True, default=[])
```

Ilustración 8: Modelos del módulo usuarios

Los servicios implementados son:

Nombre	Registro	
Descripción	Este servicio permite a los usuarios almacenar los datos personales públicos para crear una cuenta en el sistema.	
Entrada	• Certificado con el que se realiza la firma de los datos en Base64 • Datos: JSON con los datos del usuario • Cadena de caracteres de la firma de los datos en Base64	
Salida	Éxito	• 201 - Datos de usuario en formato JSON
	Error	• 400 – Bad Request • 401 - Certificado inválido • 401 - Firma inválida
Algoritmo	1. Comprobación de satisfabilidad de las entradas. En caso contrario se devuelve un error 400 (Bad Request). 2. Comprobación de Cadena de Confianza del certificado. Se comprueba si el certificado es válido dentro de los certificados de confianza del sistema. Si no es válido se devuelve un error 401 con el mensaje 'invalid_certificate'. 3. Comprobación de firma de datos. Se comprueba si los datos enviados se corresponden a los datos firmados con el certificado. En caso de no ser así se devuelve un error 401 con el mensaje 'invalid_signature'. 4. Comprobación de datos enviados. Se comprueba si los datos enviados satisfacen las restricciones del modelo de datos del usuario. En caso contrario se devuelve un error 400 (Bad Request). 5. Grabado de usuario en la base de datos. Se persisten los datos del usuario en la base de datos. 6. Retorno en formato JSON de los datos de usuario.	

Tabla 68: Módulo de usuarios: Registro

Nombre	Autenticación	
Descripción	Este servicio permite a los usuarios obtener un token de autenticación que les permitirá acceder a los demás servicios.	
Entrada	• Certificado con el que se realiza la firma de los datos en Base64 • Datos: Cadena de caracteres con el número y letra del DNI del usuario • Cadena de caracteres de la firma de los datos en Base64	
Salida	Éxito	• 200 – JSON con access token y refresh token de OAuth.
	Error	• 400 – Bad Request • 401 - Certificado inválido • 401 - Firma inválida
Algoritmo	1. Comprobación de satisfabilidad de las entradas. En caso contrario se devuelve un error 400 (Bad Request). 2. Comprobación de Cadena de Confianza del certificado. Se comprueba si el certificado es válido dentro de los certificados de confianza del sistema. Si no es válido se devuelve un error 401 con el mensaje 'invalid_certificate'. 3. Comprobación de firma de datos. Se comprueba si los datos enviados se corresponden a los datos firmados con el certificado. En caso de no ser así se devuelve un error 401 con el mensaje 'invalid_signature'. 4. Comprobación de datos enviados. Se comprueba si el DNI enviado en los datos se corresponde con el enviado en el certificado. En caso contrario se devuelve un error 400 (Bad Request). 5. Obtención de nuevo access token y refresh token de OAuth para el usuario. 6. Retorno en formato JSON del access token y del refresh token.	

Tabla 69: Módulo de usuarios: Autenticación

La siguiente infografía recoge el proceso de una manera visual:

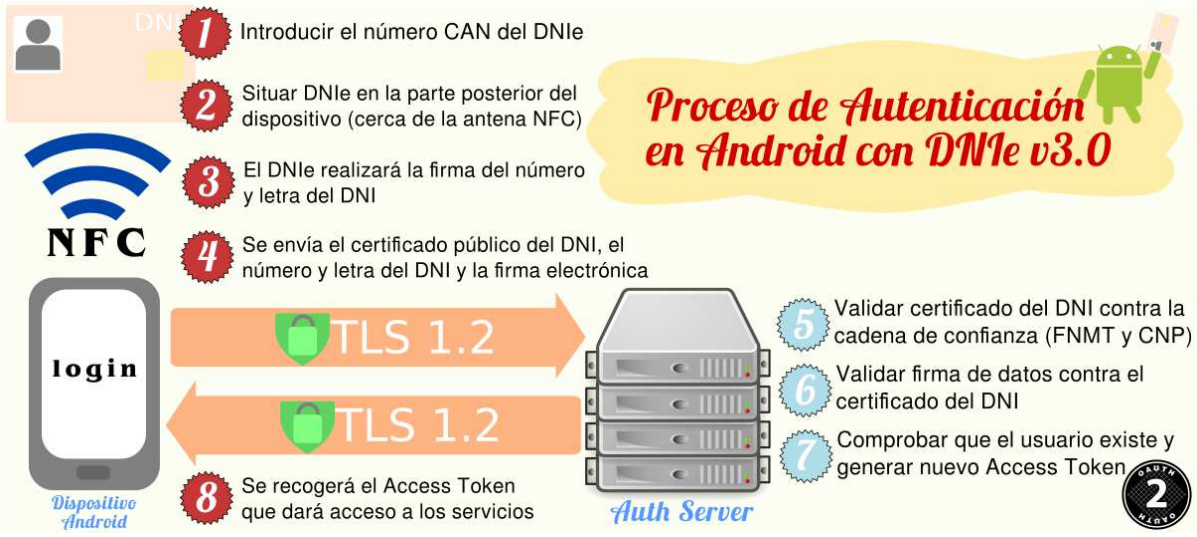


Ilustración 9: Proceso de autenticación en Android con DNIe v3.0

Nombre	Obtener datos propios	
Descripción	Este servicio permite a los usuarios obtener sus datos almacenados en el sistema.	
Entrada	• Access token	
Salida	Éxito	• 200 - Datos de usuario en formato JSON
	Error	• 401 – Not authorized
Algoritmo	1. Validación de access token. Si no es válido se retorna un error 401. 2. Retorno en formato JSON de los datos de usuario.	

Tabla 70: Módulo de usuarios: Obtener datos propios

Nombre	Modificar email	
Descripción	Este servicio permite a los usuarios modificar su email almacenado en el sistema	
Entrada	• Access token • JSON con campo email.	
Salida	Éxito	• 200 - Datos de usuario en formato JSON
	Error	• 401 – Not authorized • 400 – Bad Request
Algoritmo	1. Validación de access token. Si no es válido se retorna un error 401. 2. Validación de datos de email. Si no es un email válido se retornará un error 400. 3. Guardado de datos de usuario 4. Retorno en formato JSON de los datos de usuario.	

Tabla 71: Módulo de usuarios: Modificar email

Como se puede observar, los dos servicios que utilizan certificados para hacer operaciones no especifican que los certificados de confianza del sistema tengan que ser de la Fábrica Nacional de Moneda y Timbre, ni el certificado de entrada tenga que ser de un DNI. Esto es porque se ha decidido crear un sistema de autenticación abierto y que dependa únicamente de los certificados de confianza que requiera el sistema. Esto es que, para la aplicación que se está desarrollando en este TFG, necesitamos los certificados raíz de la Fábrica Nacional de Moneda y Timbre, así como los de las entidades certificadoras subordinadas, pero

quizás otros sistemas quieran utilizar sus propios certificados. De esta manera se deja la puerta abierta a otros desarrollos.

La validación de la cadena de confianza de los certificados, así como la validez de la firma de datos, se han hecho con la librería "crypto" de Django, la cual permite realizar de manera relativamente sencilla estas operaciones. Para ello se han tenido que descargar, en un directorio específico del proyecto, los certificados raíz públicos de la Fábrica Nacional de Moneda y Timbre y las entidades certificadoras subordinadas que expiden el DNIe.

5.1.1.2. Módulo Asambleas

Este módulo engloba toda la lógica de negocio relativa al funcionamiento de las asambleas, las propuestas y las votaciones.

Como se ha comentado antes, una asamblea es un lugar donde un grupo de personas se reúnen para proponer, debatir y votar sobre asuntos comunes. En este TFG se propone la creación de asambleas virtuales en las que las personas no tengan que reunirse físicamente.

Para dar completa funcionalidad a todo este subsistema se comenzó por crear un módulo de Django que englobara las entidades relativas al mismo.

A continuación, se implementaron las clases que definen el modelo de datos del módulo y sus relaciones tal y como define la siguiente figura.

```
class Assembly(models.Model):
    name = models.CharField(max_length=60)
    description = models.TextField()
    location = models.ForeignKey(Location)
    open = models.BooleanField(blank=True, default=False)
    elections = models.DateField(default=(datetime.now() + timedelta(days=7)).date())
    createdAt = models.DateTimeField(auto_now_add=True)
    updatedAt = models.DateTimeField(auto_now=True)

class Proposal(models.Model):
    title = models.CharField(max_length=60)
    text = models.TextField()
    assembly = models.ForeignKey(Assembly)
    deadline = models.DateField(blank=True, null=True, default=None)
    createdAt = models.DateTimeField(auto_now_add=True)
    updatedAt = models.DateTimeField(auto_now=True)

class ProposalOption(models.Model):
    title = models.CharField(max_length=30)
    proposal = models.ForeignKey(Proposal)
```

Ilustración 10: Modelos del módulo asambleas

Por cada entidad, los servicios de servidor se definieron como un RESTlike, es decir, se intenta seguir la estructura REST, pero no de manera exhaustiva. Los servicios son:

Nombre	Crear asamblea	
Descripción	Este servicio permite a los usuarios crear una asamblea	
Entrada	• Access token • JSON con datos de la asamblea.	
Salida	Éxito	• 200 - Datos de la asamblea en formato JSON
	Error	• 401 – Not authorized • 400 – Bad request
Algoritmo	1. Validación de access token. Si no es válido se retorna un error 401. 2. Validación de datos de la asamblea. En caso de error se retorna un error 400. 3. Guardado de datos de la asamblea 4. Retorno en formato JSON de los datos de la asamblea.	

Tabla 72: Módulo de asambleas: Crear asamblea

Nombre	Listar asambleas	
Descripción	Este servicio permite a los usuarios obtener una lista de las asambleas en las que es miembro	
Entrada	• Access token	
Salida	Éxito	• 200 - Datos de las asambleas en formato JSON
	Error	• 401 – Not authorized
Algoritmo	1. Validación de access token. Si no es válido se retorna un error 401. 2. Retorno en formato JSON de los datos de las asambleas.	

Tabla 73: Módulo de asambleas: Listar asambleas

Nombre	Detalle de asamblea	
Descripción	Este servicio permite a los usuarios obtener los detalles de una asamblea en la que es miembro	
Entrada	• Access token	
Salida	Éxito	• 200 - Datos de la asamblea en formato JSON
	Error	• 401 – Not authorized
Algoritmo	1. Validación de access token. Si no es válido se retorna un error 401. 2. Retorno en formato JSON de los datos de la asamblea.	

Tabla 74: Módulo de asambleas: Detalle de asamblea

Nombre	Buscar asamblea pública	
Descripción	Este servicio permite a los usuarios buscar una asamblea pública por el nombre en base a un texto.	
Entrada	• Access token • Texto de búsqueda	
Salida	Éxito	• 200 - Datos de las asambleas encontradas en formato JSON
	Error	• 401 – Not authorized
Algoritmo	1. Validación de access token. Si no es válido se retorna un error 401. 2. Retorno en formato JSON de los datos de las asambleas en cuyo título se incluya el texto de búsqueda.	

Tabla 75: Módulo de asambleas: Buscar asamblea pública

Nombre	Unirse a una asamblea pública	
Descripción	Este servicio permite a los usuarios unirse a una asamblea pública.	
Entrada	• Access token	
Salida	Éxito	• 200 - Datos de la asamblea a la que se ha unido
	Error	• 401 – Not authorized
Algoritmo	1. Validación de access token. Si no es válido se retorna un error 401. 2. Se guardan los datos del usuario como miembro de la asamblea. 3. Retorno en formato JSON de los datos de la asamblea a la que se ha unido.	

Tabla 76: Módulo de asambleas: Unirse a una asamblea pública

Nombre	Invitar a una asamblea	
Descripción	Este servicio permite a los usuarios invitar a otro usuario a una asamblea.	
Entrada	• Access token • Usuario invitado en formato JSON	
Salida	Éxito	• 200 - Datos de la asamblea a la que se ha invitado a otro usuario
	Error	• 401 – Not authorized • 400 – Bad request
Algoritmo	1. Validación de access token. Si no es válido se retorna un error 401. 2. Se comprueba la existencia del usuario invitado. Si no existe se devuelve un error 400. 3. Si es una asamblea privada y el usuario no es administrador se devolverá un error 400. 4. Se guardan los datos del invitado como miembro de la asamblea. 5. Retorno en formato JSON de los datos de la asamblea a la que se ha invitado a otro usuario.	

Tabla 77: Módulo de asambleas: Invitar a una asamblea

Nombre	Abandonar una asamblea	
Descripción	Este servicio permite a los usuarios abandonar una asamblea.	
Entrada	• Access token	
Salida	Éxito	• 200 - Datos de la asamblea a la que ha abandonado
	Error	• 401 – Not authorized
Algoritmo	1. Validación de access token. Si no es válido se retorna un error 401. 2. Se borran los datos del usuario como miembro de la asamblea. 3. Retorno en formato JSON de los datos de la asamblea que ha abandonado.	

Tabla 78: Módulo de asambleas: Abandonar una asamblea

Nombre	Crear propuesta	
Descripción	Este servicio permite a los usuarios crear una propuesta	
Entrada	• Access token • JSON con datos de la propuesta.	
Salida	Éxito	• 200 - Datos de la propuesta en formato JSON
	Error	• 401 – Not authorized • 400 – Bad request
Algoritmo	1. Validación de access token. Si no es válido se retorna un error 401. 2. Si el usuario no tiene el rol de cargo en la asamblea se devuelve en error 401 3. Validación de datos de la propuesta. En caso de error se retorna un error 400. 4. Guardado de datos de la propuesta 5. Retorno en formato JSON de los datos de la propuesta.	

Tabla 79: Módulo de asambleas: Crear propuesta

Nombre	Detalle de propuesta	
Descripción	Este servicio permite a los usuarios obtener los detalles de una propuesta en de una asamblea en la que es miembro	
Entrada	• Access token	
Salida	Éxito	• 200 - Datos de la propuesta en formato JSON
	Error	• 401 – Not authorized
Algoritmo	1. Validación de access token. Si no es válido se retorna un error 401. 2. Retorno en formato JSON de los datos de la propuesta.	

Tabla 80: Módulo de asambleas: Detalle de propuesta

Nombre	Ratificación de propuesta	
Descripción	Este servicio permite a los usuarios miembros de una asamblea con el rol de cargo aprobar o denegar la votación de una propuesta en fase de pre-propuesta.	
Entrada	• Access token • Certificado con el que se realiza la firma de los datos en Base64 • Datos: JSON con valores de aceptación o denegación • Cadena de caracteres de la firma de los datos en Base64	
Salida	Éxito	• 200 - Datos de la propuesta en formato JSON
	Error	• 401 – Not authorized
Algoritmo	1. Comprobación de satisfabilidad de las entradas. En caso contrario se devuelve un error 400 (Bad Request). 2. Validación de access token. Si no es válido se retorna un error 401. 3. Comprobación de Cadena de Confianza del certificado. Se comprueba si el certificado es válido dentro de los certificados de confianza del sistema. Si no es válido se devuelve un error 401 con el mensaje 'invalid_certificate'. 4. Comprobación de firma de datos. Se comprueba si los datos enviados se corresponden a los datos firmados con el certificado. En caso de no se así se devuelve un error 401 con el mensaje 'invalid_signature'. 5. Comprobación de datos enviados. Se comprueba si el DNI del usuario que realiza la votación se corresponde con el enviado en el certificado. En caso contrario se devuelve un error 400 (Bad Request). 6. Se guarda la opción elegida por el cargo 7. Se comprueba que la propuesta tiene los apoyos necesarios para ser votada por todos los miembros. Si es así, entonces se calcula la fecha límite de votación y se guarda como propuesta activa. 8. Retorno en formato JSON de los datos de la propuesta.	

Tabla 81: Módulo de asambleas: Ratificación de propuesta

Nombre	Votación de propuesta	
Descripción	Este servicio permite a los usuarios miembros de una asamblea votar las opciones de voto de una propuesta.	
Entrada	• Access token • Certificado con el que se realiza la firma de los datos en Base64 • Datos: JSON con la opción de voto elegida • Cadena de caracteres de la firma de los datos en Base64	
Salida	Éxito	• 200 - Datos de la propuesta en formato JSON
	Error	• 401 – Not authorized
Algoritmo	1. Comprobación de satisfabilidad de las entradas. En caso contrario se devuelve un error 400 (Bad Request). 2. Validación de access token. Si no es válido se retorna un error 401. 3. Comprobación de Cadena de Confianza del certificado. Se comprueba si el certificado es válido dentro de los certificados de confianza del sistema. Si no es válido se devuelve un error 401 con el mensaje 'invalid_certificate'. 4. Comprobación de firma de datos. Se comprueba si los datos enviados se corresponden a los datos firmados con el certificado. En caso de no ser así se devuelve un error 401 con el mensaje 'invalid_signature'. 5. Comprobación de datos enviados. Se comprueba si el DNI del usuario que realiza la votación se corresponde con el enviado en el certificado. En caso contrario se devuelve un error 400 (Bad Request). 6. Se guarda la opción elegida por el miembro 7. Retorno en formato JSON de los datos de la propuesta.	

Tabla 82: Módulo de asambleas: Votación de propuesta

5.1.1.3. Módulo Comentarios

Este módulo es dependiente de los anteriores, ya que contiene la lógica de negocio referente a los comentarios y sistema de reputación. Este módulo es muy importante, ya que uno de los objetivos buscados es que los miembros de las asambleas puedan expresar su opinión y valorar otras opiniones. Estas valoraciones conformarán el sistema de reputación de manera que una valoración positiva o negativa a un comentario revertirá en suma o resta de puntos de reputación.

La siguiente figura muestra como se ha implementado la clase perteneciente al modelo.

```
class Comment (models.Model):
    text = models.TextField()
    proposal = models.ForeignKey(Proposal)
    user = models.ForeignKey(User)
    createdAt = models.DateTimeField(auto_now_add=True)
    updatedAt = models.DateTimeField(auto_now=True)
```

Ilustración 11: Modelos del módulos de comentarios

Como podemos ver, un comentario cuenta con un atributo de texto en el que los usuarios podrán plasmar su opinión y con los atributos que relacionan el comentario a la propuesta y al usuario. Además, se añaden dos atributos de fecha de creación y actualización.

Los servicios implementados en este módulo son:

Nombre	Crear comentario	
Descripción	Este servicio permite a los usuarios miembros de una asamblea realizar un comentario a una propuesta.	
Entrada	• Access token • JSON con datos del comentario	
Salida	Éxito	• 200 - Datos del comentario en formato JSON
	Error	• 401 – Not authorized • 400 – Bad Request
Algoritmo	1. Validación de access token. Si no es válido se retorna un error 401. 2. Validación de datos del comentario. En caso de error se retorna un error 400. 3. Guardado de datos del comentario 4. Retorno en formato JSON de los datos del comentario.	

Tabla 83: Módulo de comentarios: Crear comentario

Nombre	Listar comentarios	
Descripción	Este servicio permite a los usuarios obtener una lista de los comentarios de una propuesta	
Entrada	• Access token	
Salida	Éxito	• 200 - Datos de los comentarios en formato JSON
	Error	• 401 – Not authorized
Algoritmo	1. Validación de access token. Si no es válido se retorna un error 401. 2. Retorno en formato JSON de los datos de los comentarios.	

Tabla 84: Módulo de comentarios: Listar comentarios

5.2. Cliente Android

5.2.1. Implementación

Antes de comenzar la implementación del cliente Android, se tuvo que crear un esquema a modo de *storyboard* que definiera la interfaz de usuario. La siguiente figura recoge este esquema.

Ilustración 12: Storyboard

Como se puede observar, se intentó crear una interfaz clara, concisa y siguiendo el estilo de diseño recomendado por Google. Esto responde a la necesidad de que la aplicación deberá ser accesible y adecuada a estándares que los usuarios reconozcan de otras aplicaciones.

El desarrollo de esta aplicación cliente, al igual que el servidor, se realizó por módulos.

En primer lugar, se comenzó por la parte de usuarios y autenticación mediante el DNIe en la cual se tuvieron que definir los diferentes presentadores (Activities) y modelos. La parte de autenticación llevó la mayor parte del tiempo ya que consiste en el núcleo de este TFG y conllevaba cierta dificultad.

El registro y autenticación mediante el uso del DNIe se realizó mediante la utilización de la librería para Android que provee la Oficina Técnica del DNIe, la cual permite establecer conexiones seguras con el documento. Hubo que adaptar el uso de esta librería, ya que carecía de una buena usabilidad y había que adaptarla a la forma de recolección de datos de la aplicación. Para ello, se sobrescribieron algunas clases para dotarlas de funcionalidad adicional propia. Entre estas modificaciones se encuentra el modo de obtención del número CAN y de la contraseña de acceso a los certificados. Además, se implementó una clase intermedia a modo de tarea asíncrona para poder reutilizar y lanzar en segundo plano la ejecución de la conexión y firma de datos para, de este modo, evitar bloquear el hilo de la interfaz gráfica. El proceso de autenticación y registro es el siguiente:

- Primero se pide el número CAN del DNIe para establecer contacto. Después se extraen los datos públicos del DNI con los que se construye una instancia de usuario.
- Esta instancia se transforma a formato JSON y se envía a firmar junto con el número y letra del DNI.
- Cuando termina el proceso de firma se envía la solicitud de autenticación al servidor.
    - Si el usuario ya existía y los datos enviados son correctos, se recibirá un access token.
    - En cambio, si el usuario no existía, se recibirá un error. Cuando se reciba el error de que el usuario no existe, se procederá al registro que consistirá en pedir el email y que el usuario acepte las condiciones de uso. Una vez hecho esto, se enviarán los datos de la instancia de usuario previamente firmadas para realizar su registro. Si todo va bien, a continuación, se realizará de nuevo la petición de autenticación que devolverá un access token.
- Se guardará el access token en las preferencias compartidas, de modo que el usuario no tenga que autenticarse cada vez que abra la aplicación. Con este access token ya se podrán realizar todas las llamadas a los servicios de acuerdo a las políticas implementadas en OAuth.

Este proceso permite unificar en un solo gesto la autenticación y el registro, de manera que el usuario tan solo tiene que utilizar el DNIe una sola vez para realizar ambas operaciones.

El resto de partes de la aplicación (asambleas, propuestas, comentarios) funcionan como simples clientes que se limitan a mostrar la información recibida del servicio REST, ya sea en forma de lista, de vista de detalle o de formulario de creación.

El sistema de voto dentro de la aplicación de Android funciona de forma similar a la autenticación y registro. Se utiliza la tarea asíncrona creada anteriormente para firmar los datos del voto mediante el uso del DNIe. El proceso es el siguiente:

- Primero se pide el número CAN del DNIe para establecer contacto. Después, se extraen los datos públicos del DNI con los que se construye una instancia de usuario.
- Se envía a firmar al DNIe los datos del voto en formato JSON
- Cuando termina el proceso de firma, se envía la petición de voto al servidor.
- Si la petición es correcta se recibe la información del voto y se vuelve a la vista de propuesta.

Se ha intentado que todas las operaciones que se hagan dentro de la aplicación estén bien informadas para que la experiencia de usuario sea buena y que no den lugar a confusión.

5.3. Pruebas

Este apartado recoge en detalle los diferentes casos de prueba realizados al software y sus resultados.

La realización de pruebas sirve al propósito de comprobar el correcto funcionamiento de un software. Utilizando un conjunto de casos de prueba límites se puede determinar a partir de un resultado esperado y un resultado obtenido si lo probado es válido o no.

Para formular las pruebas se utilizará la siguiente tabla:

Identificador	CP-XXX
Nombre	
Descripción	
Resultado Esperado	
Resultado Obtenido	
Resultado de prueba	[Éxito / Fracaso]

Tabla 85: Tabla base para casos de prueba

Donde:

- Identificador: Código identificador unívoco de cada caso de prueba. Será de la forma CP-XXX donde XXX serán valores enteros desde 000 hasta 999.
- Título: Texto ilustrativo del caso de prueba.
- Descripción: Texto descriptivo del caso de prueba.
- Resultado esperado: Datos o comportamiento que se espera.
- Resultado obtenido: Datos o comportamiento que se obtiene.
- Resultado de prueba: Comparación entre resultados esperados y resultados obtenidos. Puede tomar el valor de Éxito o Fracaso.

Identificador	CP-001
Título	Registro con DNIe, con todos los datos correctos
Descripción	Los usuarios deben poder registrarse si todos los datos son correctos: datos de usuario, certificado y firma.
Resultado Esperado	Se crea el usuario y se retorna un JSON con sus datos
Resultado Obtenido	Se crea el usuario y se retorna un JSON con sus datos
Resultado de prueba	Éxito

Tabla 86: CP-001, Registro con DNIe, con todos los datos correctos

Identificador	CP-002
Título	Registro con DNIe, con certificado incorrecto
Descripción	Los usuarios deben no pueden registrarse si el certificado no es válido.
Resultado Esperado	Se recibe un error de certificado no válido
Resultado Obtenido	Se recibe un error de certificado no válido
Resultado de prueba	Éxito

Tabla 87: CP-002, Registro con DNIe, con certificado incorrecto

Identificador	CP-003
Título	Registro con DNIe, con firma incorrecta
Descripción	Los usuarios deben no pueden registrarse si el certificado es válido pero la firma de los datos no es válida.
Resultado Esperado	Se recibe un error de firma no válida
Resultado Obtenido	Se recibe un error de firma no válida
Resultado de prueba	Éxito

Tabla 88: CP-003, Registro con DNIe, con firma incorrecta

Identificador	CP-004
Título	Autenticación con DNIe, con todos los datos correctos
Descripción	Los usuarios deben poder autenticarse si todos los datos son correctos: datos, certificado y firma.
Resultado Esperado	Se recibe un access token
Resultado Obtenido	Se recibe un access token
Resultado de prueba	Éxito

Tabla 89: CP-004, Autenticación con DNIe, con todos los datos correctos

Identificador	CP-005
Título	Autenticación con DNIe, con certificado incorrecto
Descripción	Los usuarios deben no pueden autenticarse si el certificado no es válido.
Resultado Esperado	Se recibe un error de certificado no válido
Resultado Obtenido	Se recibe un error de certificado no válido
Resultado de prueba	Éxito

Tabla 90: CP-005, Autenticación con DNIe, con certificado incorrecto

Identificador	CP-006
Título	Autenticación con DNIe, con firma incorrecta
Descripción	Los usuarios deben no pueden autenticarse si el certificado es válido pero la firma de los datos no es válida.
Resultado Esperado	Se recibe un error de firma no válida
Resultado Obtenido	Se recibe un error de firma no válida
Resultado de prueba	Éxito

Tabla 91: CP-006, Autenticación con DNIe, confirma incorrecta

Identificador	CP-007
Título	Obtención de datos propios con access token
Descripción	Los usuarios deben poder obtener sus datos propios si tienen un access token
Resultado Esperado	Se recibe un JSON con los datos del usuario
Resultado Obtenido	Se recibe un JSON con los datos del usuario
Resultado de prueba	Éxito

Tabla 92: CP-007, Obtención de datos propios con access token

Identificador	CP-008
Título	Obtención de datos propios sin access token
Descripción	Los usuarios no deben poder obtener sus datos propios si no tienen un access token
Resultado Esperado	Se recibe un error 401
Resultado Obtenido	Se recibe un error 401
Resultado de prueba	Éxito

Tabla 93: CP-008, Obtención de datos propios sin access token

Identificador	CP-009
Título	Modificar email con access token y datos correctos
Descripción	Los usuarios deberán poder modificar su email si tienen un access token y los datos del email son correctos
Resultado Esperado	Se reciben los datos del usuario con el email cambiado
Resultado Obtenido	Se reciben los datos del usuario con el email cambiado
Resultado de prueba	Éxito

Tabla 94: CP-009, Modificar email con access token y datos correctos

Identificador	CP-010
Título	Modificar email sin access token y datos correctos
Descripción	Los usuarios no deberán poder modificar su email si no tienen un access token, aunque los datos del email sean correctos
Resultado Esperado	Se recibe un error 401
Resultado Obtenido	Se recibe un error 401
Resultado de prueba	Éxito

Tabla 95: CP-010, Modificar email sin access token y datos correctos

Identificador	CP-011
Título	Modificar email con access token y datos incorrectos
Descripción	Los usuarios no deberán poder modificar su email si tienen un access token y los datos del email son incorrectos
Resultado Esperado	Se recibe un error 400
Resultado Obtenido	Se recibe un error 400
Resultado de prueba	Éxito

Tabla 96: CP-011, Modificar email con access token y datos incorrectos

Identificador	CP-012
Título	Crear asamblea con access token y datos correctos
Descripción	Los usuarios deberán poder crear una asamblea si tienen un access token y los datos de la asamblea son correctos
Resultado Esperado	Se crea la asamblea y se recibe un JSON con sus datos
Resultado Obtenido	Se crea la asamblea y se recibe un JSON con sus datos
Resultado de prueba	Éxito

Tabla 97: CP-012, Crear asamblea con access token y datos correctos

Identificador	CP-013
Título	Crear asamblea sin access token y datos correctos
Descripción	Los usuarios no deberán poder crear una asamblea si no tienen un access token aunque los datos de la asamblea sean correctos
Resultado Esperado	Se recibe un error 401
Resultado Obtenido	Se recibe un error 401
Resultado de prueba	Éxito

Tabla 98: CP-013, Crear asamblea sin access token y datos correctos

Identificador	CP-014
Título	Crear asamblea con access token y datos incorrectos
Descripción	Los usuarios no deberán poder crear una asamblea si tienen un access token y los datos de la asamblea son incorrectos
Resultado Esperado	Se recibe un error 400
Resultado Obtenido	Se recibe un error 400
Resultado de prueba	Éxito

Tabla 99: CP-014, Crear asamblea con access token y datos incorrectos

Identificador	CP-015
Título	Obtener lista de asambleas con access token
Descripción	Los usuarios deberán obtener una lista de las asambleas de las que es miembro si tienen un access token
Resultado Esperado	Se recibe un JSON con la lista de las asambleas a las que pertenece un usuario
Resultado Obtenido	Se recibe un JSON con la lista de las asambleas a las que pertenece un usuario
Resultado de prueba	Éxito

Tabla 100: CP-015, Obtener lista de asambleas con access token

Identificador	CP-016
Título	Obtener lista de asambleas sin access token
Descripción	Los usuarios no deberán obtener una lista de las asambleas de las que es miembro si no tienen un access token
Resultado Esperado	Se recibe un error 401
Resultado Obtenido	Se recibe un error 401
Resultado de prueba	Éxito

Tabla 101: CP-016, Obtener lista de asambleas sin access token

Identificador	CP-017
Título	Obtener detalle de asamblea con access token
Descripción	Los usuarios deberán obtener una asamblea en detalle si tienen un access token
Resultado Esperado	Se recibe un JSON con la asambleas en detalle
Resultado Obtenido	Se recibe un JSON con la asambleas en detalle
Resultado de prueba	Éxito

Tabla 102: CP-017, Obtener detalle de asamblea con access token

Identificador	CP-018
Título	Obtener detalle de asamblea sin access token
Descripción	Los usuarios no deberán obtener una asamblea en detalle si no tienen un access token
Resultado Esperado	Se recibe un error 401
Resultado Obtenido	Se recibe un error 401
Resultado de prueba	Éxito

Tabla 103: CP-018, Obtener detalle de asamblea sin access token

Identificador	CP-019
Título	Buscar asamblea pública con access token
Descripción	Los usuarios deberán obtener una lista de asambleas públicas en función de un texto de busqueda si tienen un access token
Resultado Esperado	Se recibe un JSON con la lista de las asambleas públicas encontradas o ninguna
Resultado Obtenido	Se recibe un JSON con la lista de las asambleas públicas encontradas o ninguna
Resultado de prueba	Éxito

Tabla 104: CP-019, Buscar asamblea pública con access token

Identificador	CP-020
Título	Buscar asamblea pública sin access token
Descripción	Los usuarios no deberán obtener una lista de asambleas públicas en función de un texto de busqueda si no tienen un access token
Resultado Esperado	Se recibe un error 401
Resultado Obtenido	Se recibe un error 401
Resultado de prueba	Éxito

Tabla 105: CP-020, Buscar asamblea pública sin access token

Identificador	CP-021
Título	Buscar asamblea privada
Descripción	Los usuarios no deberán obtener una lista que contenga asambleas privadas aunque se busque específicamente
Resultado Esperado	Se recibe un JSON con la lista de las asambleas públicas encontradas o ninguna
Resultado Obtenido	Se recibe un JSON con la lista de las asambleas públicas encontradas o ninguna
Resultado de prueba	Éxito

Tabla 106: CP-021, Buscar asamblea privada

Identificador	CP-022
Título	Unirse a asamblea pública
Descripción	Los usuarios podrán unirse a una asamblea pública pasando a ser miembros de la misma
Resultado Esperado	Se recibe un JSON con los datos de la asamblea
Resultado Obtenido	Se recibe un JSON con los datos de la asamblea
Resultado de prueba	Éxito

Tabla 107: CP-022, Unirse a asamblea pública

Identificador	CP-023
Título	Unirse a asamblea privada
Descripción	Los usuarios no podrán unirse a una asamblea privada
Resultado Esperado	Se recibe un error 401
Resultado Obtenido	Se recibe un error 401
Resultado de prueba	Éxito

Tabla 108: CP-023, Unirse a asamblea privada

Identificador	CP-024
Título	Invitar a una asamblea pública
Descripción	Los usuarios podrán invitar a otros usuarios a participar en una asamblea pública
Resultado Esperado	El invitado se hace miembro de la asamblea y se recibe los datos de la asamblea en formato JSON
Resultado Obtenido	El invitado se hace miembro de la asamblea y se recibe los datos de la asamblea en formato JSON
Resultado de prueba	Éxito

Tabla 109: CP-024, Invitar a asamblea pública

Identificador	CP-025
Título	Administrador invita a asamblea privada
Descripción	Los administradores de una asamblea privada podrán invitar a otros usuarios a participar en la asamblea.
Resultado Esperado	El invitado se hace miembro de la asamblea y se recibe los datos de la asamblea en formato JSON
Resultado Obtenido	El invitado se hace miembro de la asamblea y se recibe los datos de la asamblea en formato JSON
Resultado de prueba	Éxito

Tabla 110: CP-025, Administrador invita a asamblea privada

Identificador	CP-026
Título	No administrador invita a asamblea privada
Descripción	Los usuarios que no sean administradores de una asamblea privada no podrán invitar a otros usuarios a participar en la asamblea.
Resultado Esperado	Se recibe un error 401
Resultado Obtenido	Se recibe un error 401
Resultado de prueba	Éxito

Tabla 111: CP-026, No administrador invita a asamblea privada

Identificador	CP-027
Título	Abandonar una asamblea
Descripción	Los usuarios podrán abandonar una asamblea
Resultado Esperado	Se borra la membresía del usuario en la asamblea y se recibe los datos de la asamblea en formato JSON
Resultado Obtenido	Se borra la membresía del usuario en la asamblea y se recibe los datos de la asamblea en formato JSON
Resultado de prueba	Éxito

Tabla 112: CP-027, Abandonar una asamblea

Identificador	CP-028
Título	Cargo crea propuesta
Descripción	Los usuarios con el rol de cargo dentro de una asamblea podrán crear propuestas
Resultado Esperado	Se crea la propuesta y se recibe un JSON con los datos de la misma
Resultado Obtenido	Se crea la propuesta y se recibe un JSON con los datos de la misma
Resultado de prueba	Éxito

Tabla 113: CP-028, Cargo crea propuesta

Identificador	CP-029
Título	Usuario sin rol de cargo crea propuesta
Descripción	Los usuarios sin el rol de cargo dentro de una asamblea no podrán crear propuestas
Resultado Esperado	Se recibe un error 401
Resultado Obtenido	Se recibe un error 401
Resultado de prueba	Éxito

Tabla 114, Usuario sin rol de cargo crea propuesta

Identificador	CP-030
Título	Ratificación de propuesta por cargo con todos los datos de entrada correctos
Descripción	Los usuarios con el rol de cargo en una asamblea podrán ratificar o denegar la votación de una propuesta si todos los datos de entrada son correctos.
Resultado Esperado	Se crea el registro de la votación y se recibe los datos de la propuesta en formato JSON
Resultado Obtenido	Se crea el registro de la votación y se recibe los datos de la propuesta en formato JSON
Resultado de prueba	Éxito

Tabla 115: CP-030, Ratificación de propuesta por cargo con todos los datos de entrada correctos

Identificador	CP-031
Título	Ratificación de propuesta por cargo con certificado no válido
Descripción	Los usuarios con el rol de cargo en una asamblea no podrán ratificar o denegar la votación de una propuesta si el certificado no es válido
Resultado Esperado	Se recibe un error de certificado no válido
Resultado Obtenido	Se recibe un error de certificado no válido
Resultado de prueba	Éxito

Tabla 116: CP-031, Ratificación de propuesta por cargo con certificado no válido

Identificador	CP-032
Título	Ratificación de propuesta por cargo con firma incorrecta
Descripción	Los usuarios con el rol de cargo en una asamblea no podrán ratificar o denegar la votación de una propuesta si la firma de los datos no es válida
Resultado Esperado	Se recibe un error de firma no válida
Resultado Obtenido	Se recibe un error de firma no válida
Resultado de prueba	Éxito

Tabla 117: CP-032, Ratificación de propuesta por cargo con firma incorrecta

Identificador	CP-033
Título	Ratificación de propuesta por usuario no cargo
Descripción	Los usuarios que no tengan rol de cargo en una asamblea no podrán ratificar o denegar la votación de una propuesta.
Resultado Esperado	Se recibe un error 401
Resultado Obtenido	Se recibe un error 401
Resultado de prueba	Éxito

Tabla 118: CP-033, Ratificación de propuesta por usuario no cargo

Identificador	CP-034
Título	Votación de propuesta con todos los datos de entrada correctos
Descripción	Los miembros de una asamblea podrán votar una propuesta si todos los datos de entrada son correctos.
Resultado Esperado	Se crea el registro de la votación y se recibe los datos de la propuesta en formato JSON
Resultado Obtenido	Se crea el registro de la votación y se recibe los datos de la propuesta en formato JSON
Resultado de prueba	Éxito

Tabla 119: CP-034, Votación de propuesta con todos los datos de entrada correctos

Identificador	CP-035
Título	Votación de propuesta con certificado no válido
Descripción	Los miembros de una asamblea no podrán votar una propuesta si el certificado no es válido
Resultado Esperado	Se recibe un error de certificado no válido
Resultado Obtenido	Se recibe un error de certificado no válido
Resultado de prueba	Éxito

Tabla 120: CP-035, Votación de propuesta con certificado no válido

Identificador	CP-036
Título	Votación de propuesta con firma incorrecta
Descripción	Los miembros de una asamblea no podrán votar una propuesta si la firma de los datos no es válida
Resultado Esperado	Se recibe un error de firma no válida
Resultado Obtenido	Se recibe un error de firma no válida
Resultado de prueba	Éxito

Tabla 121: CP-036, Votación de propuesta con firma incorrecta

Identificador	CP-037
Título	Crear comentario
Descripción	Los usuarios miembros de una asamblea podrán crear comentarios en una propuesta
Resultado Esperado	Se crea el comentario y se reciben los datos del comentario en formato JSON
Resultado Obtenido	Se crea el comentario y se reciben los datos del comentario en formato JSON
Resultado de prueba	Éxito

Tabla 122: CP-037, Crear comentario

Identificador	CP-038
Título	Listar comentarios
Descripción	Los usuarios miembros de una asamblea podrán obtener una lista de los comentarios de una propuesta de una asamblea de la que son miembros
Resultado Esperado	Se recibe una lista de comentarios en formato JSON
Resultado Obtenido	Se recibe una lista de comentarios en formato JSON
Resultado de prueba	Éxito

Tabla 123: CP-038, Listar comentarios

5.4. Despliegue y publicación

Esta sección describe el proceso seguido para realizar el despliegue en el servidor de AWS (Amazon Web Services) y la publicación de la aplicación en la Google Play Store.

5.4.1. Despliegue en AWS

Antes de realizar el despliegue se procedió a la creación de una cuenta en la consola de Amazon para tener acceso a los servicios de AWS que permitirían más tarde la creación de una instancia de servidor.

La instancia elegida fue una t2.micro, ya que es la que se ofrece gratuitamente durante el primer año.

Además, se obtuvieron los dominios “appsamblea.es” y el subdominio “api.appsamblea.es” que fueron redirigidos hacia la IP del servidor en AWS.

Para la configuración del servidor se establecieron reglas de red por las cuales el firewall solo permitiría la interacción por los puertos 22 (ssh) y 80 (http). Esta configuración es de suma importancia ya que dota de seguridad de red al servidor evitando ataques mediante la inspección de puertos y de día cero.

Después se pasó a la instalación de los programas necesarios: Python, Pip, MySQL, Apache2 y los módulos necesarios para éste último. Además, se instaló Certbot que permite la automatización de todo lo concerniente a certificados SSL y Apache, lo que es muy importante en este proyecto ya que la seguridad de los datos enviados por la red han de estar protegidos.

La configuración de Apache se realizó de tal manera que todo el tráfico entrante se redirigiera por https obligando así a transmitir los datos por un canal seguro. Además, se crearon dos hosts virtuales de tal manera que si la dirección a la que se pretendía llegar era “appsamblea.es”, se sirve una web estática almacenada en un directorio. Pero si se pretende llegar a “api.appsamblea.es”, se sirven los servicios REST que se han descrito a lo largo de todo este documento mediante el uso de los WSGI de Python. El código de estos servicios estará almacenado en otro directorio.

Para la descarga del código de ambos directorios se utiliza un script que comprueba cada cierto tiempo si existen cambios en la rama master del repositorio Git de cada proyecto (el de la web estática y el de los servicios REST). Si existe una nueva versión, este script la descargará y la lanzará a producción.

5.4.2. Publicación en Google Play Store

Para publicar la aplicación en Google Play Store, en primer lugar, hubo que darse de alta en la consola de Google rellenando los formularios correspondientes y abonando la respectiva licencia de desarrollador.

Para poder subir un fichero instalable a Google Play Store es necesario crear una aplicación en la consola. De esta manera, se procedió a rellenar los formularios para crear la aplicación Appsamblea.

A continuación, es necesario generar un archivo APK firmado para poder subirlo. Esto se hace fácilmente con Android Studio que permite la creación de un almacén de claves y la generación de las claves necesarias para el firmado del APK.

Una vez firmado el APK, en la consola de Google, en la sección de administración de versiones se procedió a subir el APK como versión Beta abierta. La versión Beta abierta permite a cualquiera que cumpla con los requisitos del dispositivo a descargarse la aplicación.

Finalmente, solo hay que esperar a que Google actualice sus repositorios para que la aplicación aparezca en Google Play Store.

6. Metodologías usadas

Este apartado recoge las metodologías utilizadas a lo largo del análisis, diseño e implementación del sistema. Se explicará de manera breve y concisa cada metodología.

6.1. Git y gitflow

Git es un gestor de versiones de bajo nivel pensado para el mantenimiento de una gran cantidad de código distribuido entre mucha gente. Se basa en un tipo de desarrollo no lineal y descentralizado. Es el gestor de versiones más utilizado[31].

Gitflow es una metodología de trabajo con Git que permite una mejor organización de los repositorios, ramas y commits. Esta metodología establece unas reglas flexibles y adaptables para organizar el trabajo de un equipo a la hora de versionar el código. El trabajo se organiza en dos ramas principales: “master”, la cual contiene los commits del código de producción; y “develop”, que contiene los commits del código en desarrollo. Además, existen las ramas auxiliares “feature”, “release” y “hotfix” cada cual con sus reglas específicas para organizar el correcto versionado[32].

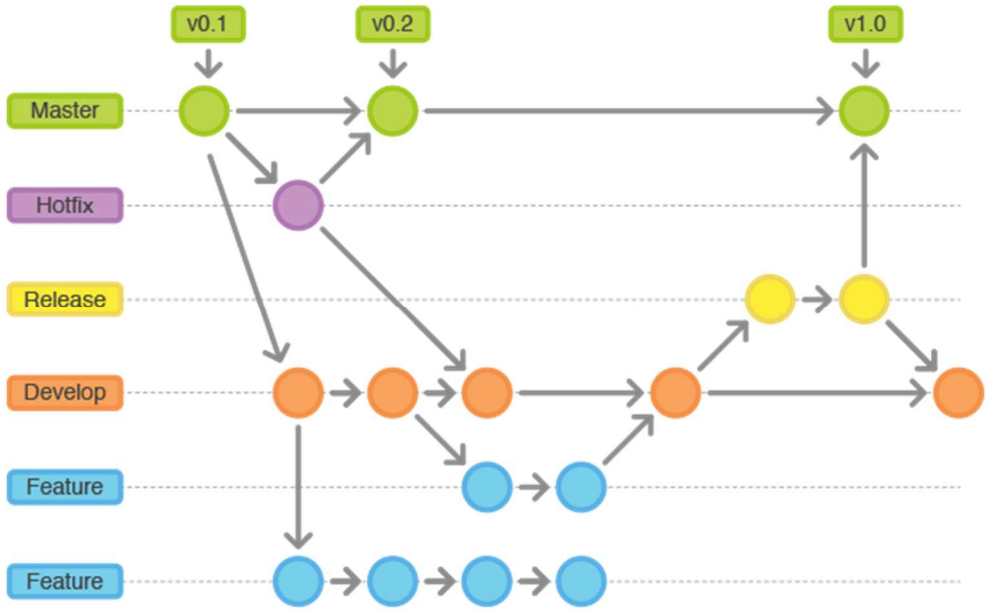


Ilustración 13: Gitflow[33]

Durante este TFG se ha seguido exhaustivamente la metodología de Gitflow y todas las reglas inherentes a cada rama. Se ha elegido esta metodología ya que es una manera ágil, muy organizada y tolerante a fallos en el versionado de código.

6.2. Integración continua

La integración continua es un modelo de automatización informática que consiste en realizar integraciones de un proyecto con bastante frecuencia. Cada cierto tiempo se descargarán los ficheros fuente desde el control de versiones (Git), se ejecutarán las pruebas y se lanzará la nueva versión [34].

En este proyecto, para realizar la integración continua no se ha contado con ningún automatizador, como Jenkins o Bamboo, sino que ha bastado con un pequeño script propio para realizar una integración continua muy básica.

6.3. Modelo vista presentador

En arquitectura de software, el patrón de diseño Modelo Vista Presentador (MVP) tiene como objetivo separar la parte de interfaz de usuario de la lógica de aplicación. Es una derivación del patrón de diseño Modelo Vista Controlador (MVC) que hace que la capa de Vista sea agnóstica en cuanto al modelo y a la lógica de la aplicación [35].

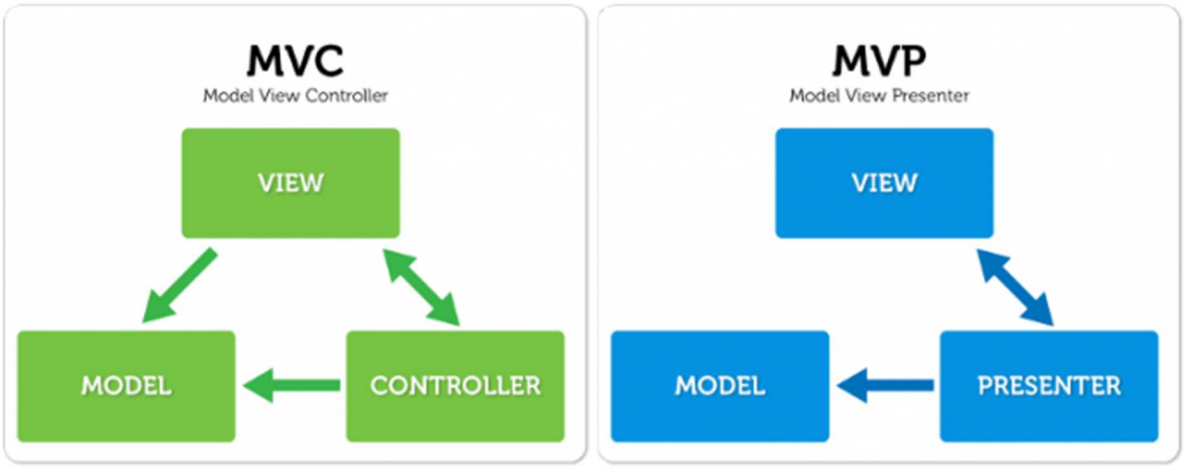


Ilustración 14: MVC vs MVP[35]

Esta arquitectura de software se ha utilizado en el TFG tan solo en la parte cliente Android. De esta manera, se ha podido controlar las vistas mediante presentadores y hacer de estas que sean reutilizables y más desacopladas de la lógica de la aplicación.

En Android, la capa de Vista pertenece a los XML y código java, que definen la interfaz de usuario; la capa de Presentación pertenece a los Activities que hacen de intermediarios entre la interfaz y la lógica de la aplicación definida por la capa de Modelo.

6.4. Modelo vista controlador

El patrón de diseño de arquitectura de software Modelo Vista Controlador, se basa en separar la lógica de negocio y los datos de la interfaz de usuario. El MVC es uno de los patrones de diseño más utilizados por su sencillez y su robustez [36].

En este TFG se ha utilizado en el servidor con el framework de Python, Django. Este framework está pensado para seguir este tipo de patrón de diseño y ofrece herramientas y documentación exhaustiva para desarrollar conforme al mismo. La capa de Vista de esta parte del proyecto la implementan los serializadores, que permiten la representación de la información como JSON. La capa de Controlador la implementan las llamadas "views" en Django y se encargan de hacer de intermediarias entre los serializadores y la capa Modelo. El framework Django incluye un ORM que sirve como abstracción del Modelo de datos y hace que las llamadas a la base de datos sean agnósticas en cuanto a tecnología, es decir, los controladores no conocen de que lenguaje ni tecnología de base de datos obtienen la información. Esto se puede configurar en los distintos ficheros de configuración de Django, que tiene soporte para multitud de tecnologías de bases de datos.

6.5. Principios SOLID

SOLID es un acrónimo mnemotécnico para representar unos principios básicos de la programación orientada a objetos con el objetivo de crear sistemas software fáciles de mantener y escalar [37].

S: Principio de responsabilidad única (Single Responsibility Principle).

O: Principio abierto/cerrado (Open/Closed Principle).

L: Principio de sustitucion de Liskov (Liskov Substitution Principle).

I: Principio de segregación de la interfaz (Interface Segregation Principle)

D: Principio de inversión de la dependencia (Dependency inversion Principle)

En este TFG se han intentado seguir estos principios a lo largo de todo el desarrollo del sistema. Estos principios permitirán que el sistema se ampliable y soporten otro tipo de tarjetas y certificados electrónicos.

6.6. OpenAPI

La especificación OpenAPI es una herramienta para desarrolladores que permite documentar, producir y consumir los servicios de un API de manera agnóstica al lenguaje que se utilice para implementarlos. Esta

especificación permite a humanos y máquinas descubrir y comprender las capacidades de los servicios sin necesidad de acceder al código, documentación o inspeccionando el tráfico de red [38], [39].

Esta herramienta soluciona el problema de documentar las APIs ya que genera de manera casi automática un documento o un sitio HTML con las especificaciones de los servicios.

En este TFG se ha utilizado este sistema de documentación para facilitar y acelerar el desarrollo del cliente. Además, ha resultado ser una buena herramienta de prueba de integración manual. La siguiente figura muestra la documentación generada de los servicios implementados.

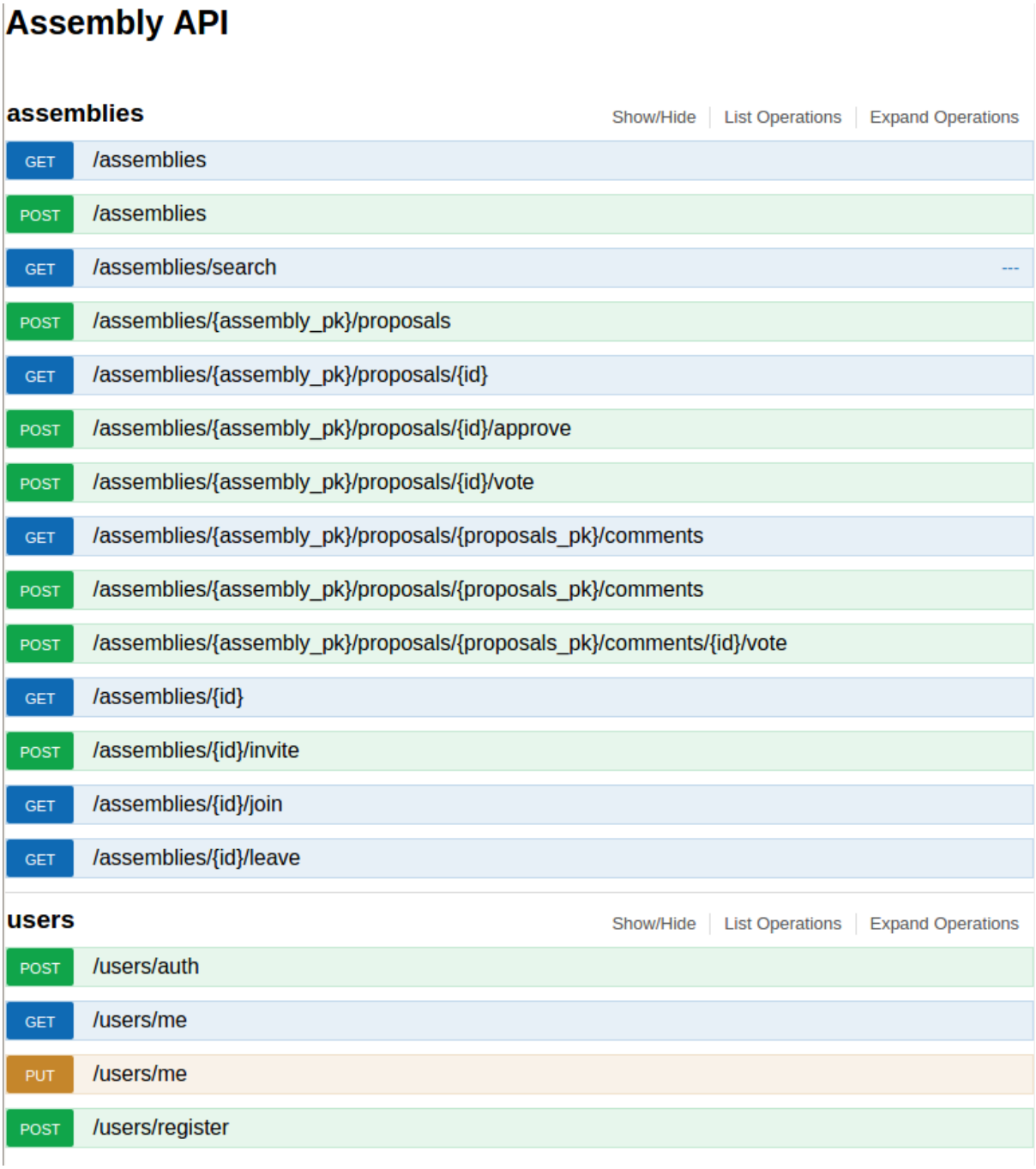


Ilustración 15: OpenAPI de Appsamblea

7. Marco regulador y aspectos legales

Este apartado recoge un breve análisis sobre las leyes y regulaciones sobre los servicios y datos que atañen a este proyecto. El marco regulador engloba las distintas normativas y legislaciones aplicables a este proyecto, desde licencias de software hasta leyes vigentes en España. Este apartado tiene especial importancia, ya que los datos que se tratan son de carácter ideológico y por tanto han de tratarse con sumo cuidado.

A lo largo del desarrollo de este TFG se han utilizado librerías externas que permitían añadir funcionalidad sin tener que desarrollarla por cuenta propia. Todas estas librerías cuentan con al menos una de las siguientes licencias de software libre: GPL, MIT, Apache o BSD. Estas licencias permiten el uso de este software de manera gratuita.

Además del software se han utilizado recursos multimedia externos extraídos de Internet, todos bajo licencia Creative Commons o de libre distribución por lo que se cumple la normativa vigente.

Según la Constitución Española en su Artículo 18.4: “La ley limitará el uso de la informática para garantizar el honor y la intimidad personal y familiar de los ciudadanos y el pleno ejercicio de sus derechos”[40].

Además de la Constitución, en España, tenemos una legislación explícita para este derecho, la Ley Orgánica de Protección de Datos de Carácter Personal (LOPD). Esta ley trata de proteger el derecho fundamental a la protección de los datos personales haciendo que los ciudadanos tengamos potestad de control sobre su uso y almacenamiento. Esta ley es de carácter obligatorio y obliga a todas las personas, empresas y organizaciones que dispongan de datos personales a cumplir unos requisitos y aplicar determinadas medidas de seguridad respecto al tipo de datos almacenados[41].

Este proyecto cumple con todo lo necesario para cumplir la LOPD, implementando las medidas de seguridad requeridas en todas las capas de la aplicación. Desde el nivel más bajo de la red, hasta el manejo y acceso a los datos de alto nivel, se sigue un riguroso control de acceso a datos. Además, los usuarios tienen que aceptar una serie de términos y condiciones de uso para acceder a la aplicación mediante la cual ceden voluntariamente sus datos para su explotación por el sistema.

Como se ha comentado antes, según la *Ley 59/2003 del 19 de diciembre de 2003,* el Documento Nacional de Identidad iguala la validez jurídica de la firma electrónica a la firma tradicional [14]. Esto supone un gran avance en la seguridad de los documentos digitales ya que permite la verificación de la identidad y otorga las cualidades de autenticación, integridad y no repudio. Con respecto a este proyecto, además, otorga validez legal a las votaciones, cosa que otras plataformas no pueden hacer. Es decir, si se hace una votación se puede verificar que los votos han sido únicos, establecer un censo de votantes y dar unos resultados legalmente válidos.

8. Análisis socioeconómico

En este apartado se hará una distinción entre dos aspectos claves de este TFG: la autenticación en sistemas mediante el uso del DNIe y los sistemas de votación. A continuación, se definirá un estudio sobre el entorno socioeconómico que puede suponer una tecnología de autenticación como esta tanto a nivel empresarial, organizacional y administrativa.

Comenzaremos por analizar el impacto del uso del DNIe como método de autenticación en sistemas. Actualmente la identificación de usuarios en sistemas se realiza principalmente mediante el uso de un nombre de usuario y una contraseña. Además de este método existen otros como el uso de tarjetas corporativas de identificación, llaves electrónicas, tokens de acceso y otros mecanismos. Todas estas formas de autenticación tienen un coste económico para toda empresa, organización o administración ya que tienen que utilizar servidores de autenticación, comprar o crear dispositivos y tarjetas para sus miembros.

El Ministerio del Interior, la Dirección General de la Policía y la Fábrica Nacional de Moneda y Timbre crearon el proyecto del DNIe para dar validez legal a la firma de los documentos digitales. Ahora, con el nuevo DNIe 3.0, se puede utilizar toda la funcionalidad de este documento identificativo por terceras personas, es decir, se abre el desarrollo de aplicaciones y servicios. Con este nuevo enfoque se pueden utilizar el DNIe como herramienta para la autenticación en sistemas y así hacer que entidades públicas y privadas ahorren en infraestructura además de mejorar su seguridad.

A parte del incentivo económico que conlleva delegar la creación de elementos identificativos y sistemas de autenticación, el uso del DNIe como identificación en sistemas implica que los ciudadanos se sentirán más propensos a su uso dado a las facilidades que conlleva. El punto fundamental que mejoraría la seguridad ciudadana en el manejo de sistemas informáticos es que, mediante el uso del DNIe los usuarios, además de tener el documento en su poder, solo tendrían que recordar una contraseña para autenticarse en servicios web. Esto evitaría las típicas libretas o post-it con credenciales de usuario, las cuales son uno de los peligros más graves a la seguridad de su información.

En cuanto a los sistemas de votación, se han analizado anteriormente los problemas actuales que tienen. El coste que conlleva validar la identidad digital de una persona puede ser excesivamente grande, por ejemplo, un sistema con cien mil usuarios en el que hubiera que validarlos a todos no habría más remedio que contratar a varias personas para realizar este proceso de una manera concienzuda, lo que conllevaría un coste económico brutal.

Mediante el uso del DNIe estos costes dejarían de existir ya que al usar los certificados electrónicos se garantiza la identidad de los usuarios. Los costes pasarían a ser los que ya tiene la administración pública al crear los documentos identificativos.

El impacto social de los sistemas de voto electrónico está patente en la sociedad, cada vez somos más y cada vez tenemos más ganas de opinar sobre los asuntos que nos rodean. Es a través de estas plataformas por las que los ciudadanos tomamos parte de estos asuntos. Queda patente que las plataformas de ideas y votaciones son algo que se demanda desde el sector de “smart-cities”, el cual propugna que los ciudadanos sean más partícipes en el desarrollo de sus ciudades y comunidades. Es por ello que una plataforma que utilice el DNIe como método de autenticación dotaría de validez legal los votos emitidos por los ciudadanos. Esta validez legal es muy necesaria ya que la principal crítica que se hace a las plataformas existentes es que pueden diferenciar si una persona tiene varias cuentas y emite más de un voto.

9. Planificación y presupuestos

La planificación de proyectos es una parte fundamental de la gestión de proyectos, en la cual, mediante cronogramas, se planea e informa del progreso. Mediante la planificación se pueden cuantificar el tiempo y los recursos necesarios para realizar un proyecto. Su objetivo es crear unas reglas con las cuales un equipo pueda ser gestionado.

Los presupuestos son necesarios en todos los proyectos y dotan de la información del coste del mismo. Persiguen plasmar detalladamente los distintos gastos que supone el desarrollo de un proyecto.

9.1. Planificación

Este apartado recoge la planificación del proyecto que se ha desarrollado a lo largo de todo este TFG. Para diseñar esta planificación se han utilizado diagramas Gantt. Un diagrama Gantt es una herramienta gráfica que trata de mostrar el tiempo previsto para cada una de las tareas de un proyecto dentro de un marco temporal fijo. Este diagrama deberá ser consultado y revisado según avance el proyecto.

Como los diagramas Gantt necesitan tareas para después poder estimar su duración, en primer lugar, se habrán de crear dichas tareas. Las tareas de este proyecto son:

- Definición del problema
  - Lectura de artículos divulgativos.
  - Revisión de soluciones existentes.
- Investigación
  - Investigación de tecnología del DNIe.
  - Investigación de soluciones con DNIe.
  - Investigación de tecnología NFC en Android.
  - Investigación de firma con DNIe mediante NFC en Android.
- Formación
  - Aprendizaje y prueba de framework Django.
  - Aprendizaje y prueba de Android SDK.
  - Aprendizaje y prueba de librería de DNIe para Android.
- Definición de objetivos

- Comparación y documentación de soluciones existentes
- Análisis de la solución
  - Casos de uso.
  - Requisitos de software.
- Diseño de la solución
  - Diseño de clases y relaciones.
  - Diseño de componentes.
  - Diseño del entorno operacional.
- Implementación de la solución
  - Implementación de módulo de usuarios en Django.
  - Implementación de módulo de asambleas en Django.
  - Implementación de módulo de comentarios en Django.
  - Implementación de módulo de usuarios en Android.
  - Implementación de módulo de asambleas en Android.
  - Implementación de módulo de comentarios en Android.
- Pruebas de la solución
- Documentación del proyecto

Nombre	Fecha d...	Fecha ...
Definición del problema	1/03/17	10/03/17
Lectura de artículos divulgativos	1/03/17	10/03/17
Revisión de soluciones existentes	1/03/17	10/03/17
Investigación	8/03/17	24/03/17
Investigación de tecnología del DNIe	8/03/17	14/03/17
Investigación de soluciones con DNIe	8/03/17	14/03/17
Investigación de tecnología NFC en Android	13/03/17	17/03/17
Investigación de firma con DNIe mediante NFC en Android	13/03/17	24/03/17
Formación	20/03/17	31/03/17
Aprendizaje y prueba de framework Django	20/03/17	24/03/17
Aprendizaje y prueba de Android SDK	20/03/17	24/03/17
Aprendizaje y prueba de librería de DNIe para Android	20/03/17	31/03/17
Definición de objetivos	6/03/17	31/03/17
Comparación y documentación de soluciones existentes	6/03/17	31/03/17
Análisis de la solución	3/04/17	14/04/17
Casos de uso	3/04/17	14/04/17
Requisitos de software	3/04/17	14/04/17
Diseño de la solución	10/04/17	21/04/17
Diseño de clases y relaciones	10/04/17	21/04/17
Diseño de componentes	11/04/17	21/04/17
Diseño del entorno operacional	17/04/17	21/04/17
Implementación de la solución	10/04/17	2/06/17
Implementación de módulo de usuarios en Django	10/04/17	21/04/17
Implementación de módulo de asambleas en Django	24/04/17	28/04/17
Implementación de módulo de comentarios en Django	1/05/17	5/05/17
Implementación de módulo de usuarios en Android	8/05/17	19/05/17
Implementación de módulo de asambleas en Android	22/05/17	26/05/17
Implementación de módulo de comentarios en Android	29/05/17	2/06/17
Pruebas de la solución	24/04/17	9/06/17
Documentación del proyecto	1/03/17	9/06/17

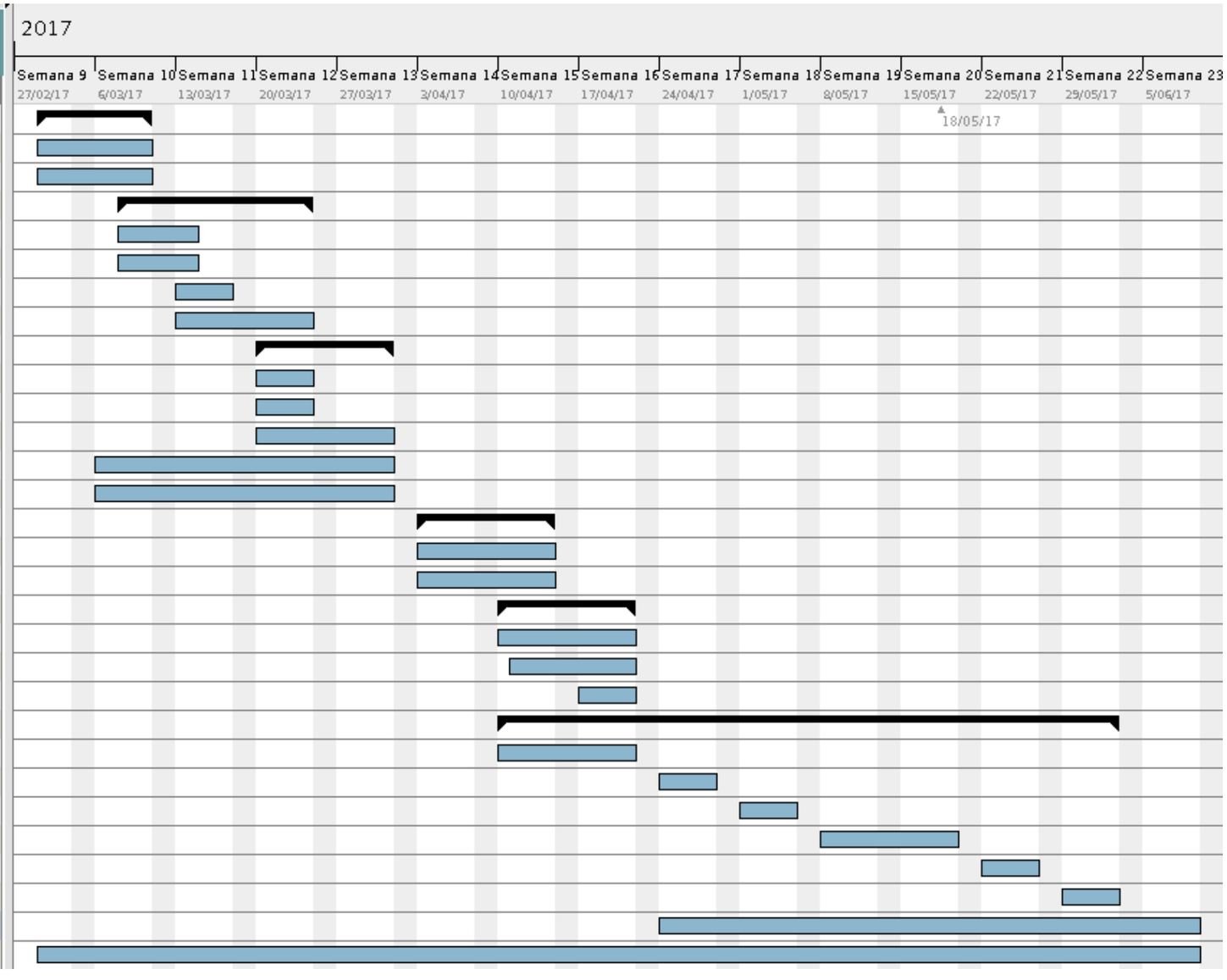


Ilustración 16: Planificación, Diagrama de Gantt

9.2. Presupuesto

Este presupuesto tratará de estimar el coste total del proyecto a partir de los datos recabados en la planificación anterior junto con los gastos en material y costes derivados del desarrollo.

9.2.1. Gastos de personal

El personal es el gasto que más recursos consume a la hora de estimar los costes en un proyecto. El personal implicado en la realización de este TFG ha sido:

- Raúl Bejarano Parrilla: Analista, diseñador y desarrollador del proyecto. Encargado de realizar la documentación entregable del proyecto.
- José María Álvarez Rodríguez: Tutor del proyecto. Encargado de solucionar dudas y aportar consejos durante el desarrollo del proyecto.

La siguiente tabla recoge los costes de personal en conformidad con la planificación del apartado anterior:

Persona	Función	Coste(€/hora)	Horas empleadas	Coste por persona
Raúl Bejarano Parrilla	Analista, dis	20,00 €	800,00 €	16.000,00 €
José María Álvarez Rodríguez	Tutor	30,00 €	30,00 €	900,00 €
			Total	16.900,00 €

Tabla 124: Gastos de personal

El coste por hora de Raúl Bejarano Parrilla y José María Álvarez Rodríguez se ha estimado en función de sus competencias y el mercado laboral en su sector.

9.2.2. Gastos de material

Los gastos materiales engloban los costes de los objetos, dispositivos y recursos físicos que se han utilizado a lo largo del proyecto.

Equipos informáticos

Los costes asociados a los equipos informáticos serán estimados según su uso y su estimación de amortización, en este caso 4 años (48 meses). La siguiente tabla recoge estos gastos:

Equipo	Cantidad	Coste und. (€)	Amortización (meses)	Uso (meses)	Total (€)
Dell XPS 13	1	1.223,00 €	48	3	76,44 €
Oneplus One	1	299,00 €	48	3	18,69 €
				Total	95,13 €

Tabla 125: Gastos de equipos informáticos

Impresión de documentación

Los gastos asociados a los gastos de impresión son estimados en función del precio del coste unitario de la impresión a color de una cara de un folio en tamaño DIN A4 y la estimación en número de páginas de la documentación. La siguiente tabla recoge estos gastos:

Concepto	Cantidad	Coste und. (€)	Número de copias	Total (€)
Hojas a color	120	0,50 €	3	180,00 €
			Total	180,00 €

Tabla 126: Gastos de impresión

9.2.3. Gastos de licencias

Los gastos de licencia hacen referencia a las licencias de software o permisos que se han necesitado a lo largo del desarrollo del proyecto. La siguiente tabla recoge estos gastos:

Licencia	Cantidad	Coste und. (€)	Amortización (meses)	Uso (meses)	Total (€)
Microsoft Office 2016	1	254,00 €	24	3	31,75 €
Desarrollador de Google	1	22,00 €	24	3	2,75 €
				Total	34,50 €

Tabla 127: Gastos de licencias

9.2.4. Gastos indirectos

Este apartado detalla los costes que indirectamente afectan al proyecto. La siguiente tabla recoge estos gastos

Concepto	Cantidad	Coste mensual. (€)	Número de meses	Total (€)
Conexión a Internet	1	40,00 €	3	120,00 €
Electricidad y agua	1	5,00 €	3	15,00 €
			Total	135,00 €

Tabla 128: Gastos indirectos

9.2.5. Gasto total del proyecto

El gasto total del proyecto engloba todos los gastos anteriores y conforma el coste íntegro de la realización del proyecto. La siguiente tabla recoge estos gastos:

Concepto	Total (€)	Total (%)
Gastos de personal	16.900,00 €	67,24%
Gastos de material	95,13 €	0,38%
Gastos de licencias	180,00 €	0,72%
Gastos indirectos	135,00 €	0,54%
Ganancia esperada	3.462,03 €	13,77%
IVA	4.362,15 €	17,36%
Total	25.134,30 €	100%

Tabla 129: Gasto total del proyecto

10. Conclusiones y Trabajo Futuro

La identidad digital es un problema para muchas organizaciones públicas y privadas que quieren que sus usuarios actúen de manera unívoca en su sistema. En España contamos con la ventaja de un organismo del Estado que expide documentos de identificación con certificados digitales que se pueden aprovechar para solventar este problema. La utilización de estos documentos, junto con los sistemas de autenticación vigentes como OAuth2, es una necesidad patente. Mediante la integración que propone este TFG, es posible identificar a una persona de manera única en sistemas conectados.

Es visible que cada día que pasa los ciudadanos queremos ser más partícipes de la vida política de nuestra ciudad y formar parte de la toma de decisiones en nuestras comunidades. Es por ello por lo que se empiezan a utilizar plataformas de participación ciudadana en muchas de nuestras ciudades. Es esencial, por tanto, tener sistemas de votación abiertos, que fomenten la participación de todos, sin discriminaciones en su usabilidad y que encuentren sinergias con tecnologías de uso cotidiano. Estas plataformas deben contar con una seguridad especial ya que usan datos sensibles de carácter ideológico. Además, es prioritario que estas plataformas tengan sistemas de verificación del voto para garantizar que una persona solo pueda votar una vez.

El uso del DNIe a nivel empresarial aún está estancado debido al desconocimiento de estas organizaciones de los grandes beneficios que les puede otorgar. A lo largo del desarrollo de este TFG se han podido ver diferentes aplicaciones de terceros que ya utilizan esta tecnología. Mi opinión personal es que es una tecnología muy potente y a punto de resurgir, pero necesita un pequeño impulso de adopción por parte de las empresas tecnológicas.

Actualmente el desarrollo solo es estable en Android con el DNIe v3.0, aunque se está trabajando en una versión de cliente web que tenga la misma funcionalidad con la misma seguridad y que sirva para cualquier versión del DNI electrónico. Este paso es necesario ya que las restricciones tecnológicas de la versión actual limitan mucho su uso por parte de los usuarios.

Además, para evitar la centralización de datos, en un futuro próximo se migrará toda la parte de servidor a un entorno distribuido Blockchain. Este entorno permitirá mejorar la seguridad de los datos además de mejorar la auditoría de los votos por parte de entidades externas. Como Blockchain permite que sus nodos no tengan confianza entre ellos, cualquier interesado en mejorar las capacidades del sistema utilizando sus sistemas, será posible aumentar la potencia sin aumentar los costes.

Para terminar, he de analizar los beneficios personales que me ha aportado el desarrollo de este trabajo. En primer lugar, y como dije al principio, tenía este proyecto en mente desde hace tiempo, pero por causas personales, de tiempo y de tecnología (aún no existía una implementación NFC para Android) no pude desarrollarlo. Es con este TFG con el que he podido empezar a esbozar un proyecto de largo recorrido y del que espero que más gente se sume después de realizar esta pequeña prueba de concepto. Enseñando el

resultado final de mi trabajo, me he dado cuenta de que la gente demanda este tipo de iniciativas y por ello seguiré desarrollando en un futuro este proyecto para que sea lo más útil posible a todos esos ciudadanos que tienen ganas de participar y mejorar su comunidad.

11. Glosario

Access token	Código alfanumérico necesario para realizar un acceso a los datos de un sistema.
API	Siglas de Aplication Program Interface. Conjunto se subrutinas, funciones y procedimientos ofrecidos para ser utilizados por otro software.
Apk (fichero)	Siglas de Android Package Kit.
Backend	Parte de un sistema que se aloja en un servidor. Suele encapsular la lógica de negocio.
Blockchain	Base de datos distribuida formada por cadenas de bloques.
Bluetooth	Especificación industrial para redes inalámbricas personales.
Ciber-delincuente	Persona que utiliza las nuevas tecnologías para cometer delitos.
Crowdsourcing	Modelo colaborativo en el cual los individuos u organizaciones utilizan las nuevas tecnologías para obtener servicios o ideas.
DNI	Siglas de Documento Nacional de Identidad
DNIe	Siglas de Documento Nacional de Identidad electrónico
Framework	Conjunto estandarizado de conceptos, prácticas y criterios para enfocar un problema.
Gamificación	Uso de técnicas, elementos y dinámicas de los juegos para reforzar conductas y promover la participación.
Infografía	Representación visual informativa
Javascript	Lenguaje de programación interpretado.
JSON	Siglas de Javascript Object Notation. Formato de texto ligero para el intercambio de datos.
Librería (software)	Conjunto de funcioalidades de software con una interfaz definida.
NFC	Siglas de Near Field Communication
ORM	Siglas de Object Related Manager.
REST	Siglas de Representational State Transfer.
Script	Conjunto de instrucciones ejecutables en un sistema.
Serializador	Herramienta software que permite cambiar el tipo de representación de datos.
Smart card	Tarjetas que contienen chips electrónicos
smart-city	Concepto para definir la adecuación de la tecnología en entorno de una población.
Smart-contract	Programa informático de las redes Blockchain
Smartphone	Tipo de teléfono móvil con las características de un computador

Software	Conjunto de programas y rutinas que permiten a un ordenador realizar tareas específicas
SSL	Siglas de Secure Socket Layer.
Storyboard	Guión gráfico que sirve para explicar un tema.
T&C	Siglas de Términos y Condiciones.
TFG	Siglas de Trabajo de Fin de grado
TPV	Siglas de Terminal Punto de Venta.
URL	Siglas de Uniform Resource Locator.
Wireless	Inalámbrico.
WSGI	Siglas de Web Server Gateway Interface
XML	Siglas de eXtensible Markup Language.

12. Bibliografía

[1] X. O. source semantic web C. for X.- http://www.ximdex.com, «Agencia Española de Protección de Datos». [En línea]. Disponible en: http://www.agpd.es/portalwebAGPD/canaldocumentacion/informes_juridicos/reglamento_lopd/index-ides-idphp.php. [Accedido: 08-jun-2017].

[2] «Identidad 2.0», *Wikipedia, la enciclopedia libre*. 26-mar-2017.

[3] «Identidad digital | t-Form@s». [En línea]. Disponible en: http://www2.ual.es/ci2bual/comunicar-la-informacion/identidad-digital/. [Accedido: 12-jun-2017].

[4] «OpenID», *Wikipedia, la enciclopedia libre*. 18-may-2017.

[5] «Por qué no puedes votar en internet con el DNI electrónico», *EL ESPAÑOL*, 22-may-2015. [En línea]. Disponible en: http://blog.elespanol.com/actualidad/por-que-no-puedes-votar-en-internet-con-el-dni-electronico/. [Accedido: 08-jun-2017].

[6] «¿Qué es el voto electrónico?», *Blog de Javier Smaldone*, 26-mar-2017. [En línea]. Disponible en: https://blog.smaldone.com.ar/2017/03/26/que-es-el-voto-electronico/. [Accedido: 12-jun-2017].

[7] P. J. V|julio 12th y 2012|Blog|Comentarios desactivados en Voto Electrónico-Voto presencial, «Voto Electrónico – Voto presencial | Custom Vote». .

[8] E. & Pablo (urjc), «DEMOCRACIA ELECTRÓNICA: TIPOS DE VOTO ELECTRÓNICO.. ¿CUÁL ES MEJOR?», *DEMOCRACIA ELECTRÓNICA*, 13-feb-2013. .

[9] W. J. Kelleher, «Internet Voting: The Great Security Scare», 2009.

[10] «Ventajas y desventajas del voto electrónico», *Ventajas y desventajas del voto electrónico ~ GigaTecno - Blog de Tecnología*. .

[11] D. M. Laura, «Blogfolio de Deres Maria Laura: VENTAJAS Y DESVENTAJAS DEL VOTO ELECTRONICO», *Blogfolio de Deres Maria Laura*, viernes, de agosto de-2011. .

[12] «DNI electrónico en España», *Wikipedia, la enciclopedia libre*. 31-may-2017.

[13] «Descripción del Chip DNIe 3.0». [En línea]. Disponible en: https://www.dnielectronico.es/PortalDNIe/PRF1_Cons02.action?pag=REF_1078&id_menu=[26_%2030]. [Accedido: 12-jun-2017].

[14] «BOE.es - Documento BOE-A-2003-23399». [En línea]. Disponible en: https://www.boe.es/buscar/doc.php?id=BOE-A-2003-23399. [Accedido: 12-jun-2017].

[15] «Firma electrónica y DNI 3.0 | Firma-e». .

[16] J. Penalva, «NFC: qué es y para qué sirve», *Xataka*, 25-ene-2011. [En línea]. Disponible en: https://www.xataka.com/moviles/nfc-que-es-y-para-que-sirve. [Accedido: 08-jun-2017].

[17] «Near field communication», *Wikipedia, la enciclopedia libre*. 04-jun-2017.

[18] E. por: P. V. 2016-12-21T14:00:00Z H. 5 meses, «NFC: ¿Qué es y para qué sirve?», *AndroidPIT*. [En línea]. Disponible en: http://www.androidpit.es/nfc-que-es-para-que-sirve. [Accedido: 12-jun-2017].

[19] «Innovation Management Software», *IdeaScale*, 02-sep-2014. [En línea]. Disponible en: https://ideascale.com/. [Accedido: 08-jun-2017].

[20] «IdeaScale», *Wikipedia*. 29-may-2017.

[21] «Agora Voting - Home». [En línea]. Disponible en: https://agoravoting.com/. [Accedido: 08-jun-2017].

[22] «Agora Voting», *Wikipedia, la enciclopedia libre*. 11-may-2017.

[23] J. F. Valencia, «Podemos y Compromís, primeros clientes de la plataforma de voto Agora Voting», *heraldo.es*, 03-mar-2015. [En línea]. Disponible en: http://www.heraldo.es/noticias/nacional/2015/03/03/podemos_compromis_primeros_clientes_plataforma_voto_agora_voting_343268_305.html. [Accedido: 08-jun-2017].

[24] «Agora Voting - Overview». [En línea]. Disponible en: https://agoravoting.com/overview/. [Accedido: 08-jun-2017].

[25] «Change.org», *Wikipedia, la enciclopedia libre*. 05-jun-2017.

[26] M. Saavedra, «Seis euros por número de móvil, 1,5 por e-mail: el lucrativo negocio de Change.org», *Vozpópuli*, 10-jul-2015. [En línea]. Disponible en: http://www.vozpopuli.com/economia-y-finanzas/empresas/change-org-Peticiones-Data_mining-Change-org-Avaaz-Peticiones_online-data_mining_0_823717647.html. [Accedido: 08-jun-2017].

[27] «Decide Madrid: portal de participación ciudadana de Madrid». [En línea]. Disponible en: https://decide.madrid.es/. [Accedido: 08-jun-2017].

[28] «CONSUL | Gobierno abierto y Participación ciudadana». [En línea]. Disponible en: http://www.decide.es/es/. [Accedido: 08-jun-2017].

[29] «GitHub - consul/consul: Consul - Open Government and E-Participation Web Software». [En línea]. Disponible en: https://github.com/consul/consul. [Accedido: 08-jun-2017].

[30] C. Ochoa, «Muestreo probabilístico: muestreo aleatorio simple». [En línea]. Disponible en: https://www.netquest.com/blog/es/blog/es/muestreo-probabilistico-muestreo-aleatorio-simple. [Accedido: 14-jun-2017].

[31] «Git», *Wikipedia, la enciclopedia libre*. 13-may-2017.

[32] «A successful Git branching model», *nvie.com*. [En línea]. Disponible en: http://nvie.com/posts/a-successful-git-branching-model/. [Accedido: 14-jun-2017].

[33] «Read Git Flow | Leanpub». [En línea]. Disponible en: https://leanpub.com/git-flow/read. [Accedido: 15-jun-2017].

[34] «Integración continua», *Wikipedia, la enciclopedia libre*. 30-nov-2016.

[35] «Modelo Vista Presentador (MVP) en Android», *Develapps*. [En línea]. Disponible en: http://www.develapps.com/es/noticias/modelo-vista-presentador-mvp-en-android. [Accedido: 08-jun-2017].

[36] «Modelo–vista–controlador», *Wikipedia, la enciclopedia libre*. 07-jun-2017.

[37] S. Oloruntoba, «S.O.L.I.D: The First 5 Principles of Object Oriented Design», *Scotch*. [En línea]. Disponible en: https://scotch.io/bar-talk/s-o-l-i-d-the-first-five-principles-of-object-oriented-design. [Accedido: 14-jun-2017].

[38] «GitHub - OAI/OpenAPI-Specification: The OpenAPI Specification Repository». [En línea]. Disponible en: https://github.com/OAI/OpenAPI-Specification. [Accedido: 14-jun-2017].

[39] «Repo», *Open API Initiative*. .

[40] «Título I. De los derechos y deberes fundamentales - Constitución Española». [En línea]. Disponible en: http://www.congreso.es/consti/constitucion/indice/titulos/articulos.jsp?ini=18&tipo=2. [Accedido: 14-jun-2017].

[41] «Ley Orgánica de Protección de Datos de Carácter Personal (España)», *Wikipedia, la enciclopedia libre*. 17-abr-2017.